ОНТОЛОГИИ И ТЕЗАУРУСЫ

Учебное пособие

Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В.

Казань, Москва
2006

АННОТАЦИЯ

Предлагаемый курс направлен на формирование базовых знаний об онтологиях и тезаурусах и практических навыков по проектированию и применению онтологий при разработке компонентов интеллектуального программного обеспечения. Курс знакомит студентов с основными понятиями области представления знаний, примерами лучшего опыта в разработке систем основанных на онтологиях и тезаурусах, описывает задачи, решаемые с их помощью, а также набор инструментальных средств проектирования и представления онтологий и информационно-поисковых тезаурусов.

Центральное место в курсе отводится тематике информационного поиска. Подробно рассматриваются как традиционные методы поиска: по ключевым словам, основанные на техниках двоичного поиска, ранжированного поиска и вероятностных моделях, так и подходы к улучшению качества поиска с помощью тезаурусов и онтологий. В части лекций, посвященных проектированию онтологий, наравне с описанием классических и современных методологий внимание уделяется разбору примеров реальных проектов.

Данное пособие предназначено для пояснения основных положений материалов лекций (в виде слайдов презентаций), которые являются основной частью курса.

Курс разработан в Российском научно-образовательном центре по лингвистике им. И.А.Бодуэна де Куртенэ в рамках программы создания серии инновационных курсов.

В настоящее время курс читается на факультете ВМиК Казанского государственного университета.

Разные фрагменты курса апробировались ранее в курсах, читавшихся в Казанском государственном университете:

- В.Д.Соловьевым на ф-те ВМК осенью 2005 г. “Обработка естественного языка on-line”;
- В.Д.Соловьевым на ф-те ВМК осенью 2004 г. “Информационный поиск, категоризация текстов, автоматическое резюмирование”

и в Московском государственном университете:

- Н.В.Лукашевич на филологическом факультете весной 2003 г. "Автоматическая обработка больших текстовых коллекций";

- Н.В.Лукашевич на филологическом факультете осенью 2003 г. "Лингвистические онтологии для автоматической обработки текстов";
- Б.В.Добровым на факультете ВМиК осенью 2004 г. "Интеллектуальные информационные технологии (знания и машинное обучение в информационном поиске)";
- Н.В.Лукашевич на факультете ВМиК весной 2005 г. "Онтологии и автоматическая обработка текстов".

Данный курс также основан на материале лекций, прочитанных Н.В.Лукашевич на Казанских школах по компьютерной лингвистике в 2001-2004 гг.

Предлагаемый курс был назван победителем:

- открытого конкурса учебных курсов в области разработки программного обеспечения, организованного компанией Microsoft и факультетом вычислительной математики и кибернетики МГУ им.М.В.Ломоносова в 2006 году;
- конкурса учебных курсов по информационному поиску «Класс 2006», организованного компанией Яндекс.

Авторы надеются, что курс будет способствовать привлечению талантливой молодежи в область информационного поиска и появлению специализированных средств и библиотек для применения онтологий в этой сфере.

Базовые требования к слушателям курса ограничиваются знаниями по математике и компьютерным наукам в объеме программы начальных курсов университетов.

СОДЕРЖАНИЕ

- В написании Темы 1 пособия принимали участие все авторы.
- Темы 2 и 3 написаны В.Д.Соловьевым и В.В.Ивановым при участии Н.В.Лукашевич. Тема 5 написана этими авторами совместно.
- Темы 4, 6 и 7 написаны В.Д.Соловьевым и В.В.Ивановым.
- Темы 8-10 написаны Н.В.Лукашевич и Б.В.Добровым.

Рекомендуемая литература

1. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. - С-Пб.: Питер, 2000. - 384 с.
2. Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий // Электронные библиотеки – Москва: Институт развития информационного общества, т.7 вып.4, 2004. – Электронный журнал, посвященный созданию и использованию электронных библиотек. - (Рус.). - URL: http://www.elbib/.
3. Разработка онтологий 101: руководство по созданию Вашей первой онтологии. - ifets.ieee.org/russian/depository/ontology101_rus.doc
4. http://www.w3c.org/TR/2004/REC-owl-guide-20040210/ (в Интернете доступно русскоязычное описание языка OWL)
5. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller Introduction to WordNet: An On-line Lexical Database.
6. George A. Miller Nouns in WordNet: A Lexical Inheritance System.
7. Lars Marius Garshol Metadata? Thesauri? Taxonomies? Topic Maps! Making sense of it all. (http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html)
8. Topic Maps – A Standard For Information Organisation (http://www.techquila.com/ topicmaps.html)
9. Berners-Lee, T., Hendler J., & Lassila, O. (2001). The semantic web. Scientific American, 284(5), 34-43.
10. N. Fridman Noy and C.D. Hafner: The State of the Art in Ontology Design, AI Magazine, 18(3):53---74, 1997. (http://www.aaai.org/Library/Magazine/Vol18/18-03/Papers/AIMag18-03-005.pdf)
11. Uschold M., Gruninger M. Ontologies: Principles, Methods and Applications. In *Knowledge Engineering Review* 11(2), 1996, pp. 93–155.
12. Gruber T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *KSL-93-04, Knowledge Systems Laboratory*, Stanford University, 1993.
13. Miller, George A., Christiane Fellbaum, Judy Kegl and Katherine J. Miller. WordNet: an electronic lexical reference system based on theories of lexical memory. In: *Revue quebecoise de linguistique* 17 (2), 1988, pp. 181 - 213.

Полезные ссылки

- http://www.dialog-21.ru/
- http://swoogle.umbc.edu/
- http://www.w3.org/2001/sw/WebOnt/.
- http://wonderweb.semanticweb.org/.
- http://www.w3.org/2001/sw/BestPractices/.
- http://www.xml.com/pub/a/2002/11/06/ontologies.html.

1. ВВЕДЕНИЕ. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Слово «онтология» имеет два значения**:**

- **Онтология 1. -** •Философская дисциплина, которая изучает наиболее общие характеристики бытия и сущностей;
- **Онтология 2.** – артефакт, структура, описывающая значения элементов некоторой системы.

Курс посвящен способам разработки и использования в приложениях онтологий как артефактов (Онтология 2).

Неформально, онтология представляет собой некоторое описание взгляда на мир применительно к конкретной области интересов. Это описание состоит из терминов и правил использования этих терминов, ограничивающих их значения в рамках конкретной области

На формальном уровне, онтология - это система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно строить классы, объекты, отношения, функции и теории.

Основными компонентами онтологии являются:

- Классы или понятия,
- Отношения,
- Функции,
- Аксиомы,
- Примеры.

Одно из самых известных определений онтологии, сформулированное Грубером таково: Онтология – это спецификация концептуализации.

Концептуализация – это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации.

Например, если мы рассматриваем простую предметную область, описывающую кубики на столе, то концептуализацией является набор возможных положений кубиков, а не конкретное их расположение в текущий момент времени.

В качестве примера того, что в рамках онтологий понимается под аксиомами, можно привести следующее положение и его формальную запись на языке исчисления предикатов первого порядка:

Работник, являющийся руководителем проекта, работает в проекте

Вводятся переменные *E* (работник), *P* (руководитель проекта):

```
Forall (E,P) Employee(E) and Head-Of-Project(E,P)
=> Works-At-Project(E,P)
```

В данном курсе будут рассмотрены следующие вопросы:

- существующие классификации онтологий по разным основаниям,
  - отношение понятий онтологий и лексических значений; существующие лингвистические (лексические) онтологии
  - применение онтологии в решении различных задач, в частности:
  - онтологии в концепции Semantic Web;
  - онтологии для решения задач информационного поиска;
  - онтологии для интеграции разнородных источников данных;
- структура конкретных онтологий таких как
  - онтологии верхнего уровня,
  - онтология вина и пищи,
  - онтология в сфере культурного наследия CIDOC CRM,
- структура, проблемы и применение наиболее известной лингвистической онтологии WordNet;
- традиции использования таких ресурсов для информационного поиска как информационно-поисковые тезаурусы, которые рассматриваются как вид онтологических ресурсов, и методы их использования в современных условиях, характеризующееся значительным преобладанием автоматических режимов обработки текста
- принципы разработки специальных тезаурусов как ресурсов для автоматической обработки текстов, которые соединяют в себе три существующие традиции в области разработки ресурсов онтологического типа: формальные онтологии, лингвистические онтологии, традиционные информационно-поисковые тезаурусы, а также использование такого тезауруса для автоматической обработки текстов в различных приложениях в области информационного поиска.

Вопросы к лекции

1. Что такое онтология?
2. Составные части онтологий

Литература

1. Гаврилова Т., Хорошевский В. Базы знаний интеллектуальных систем. – Питер, 2002
2. Тим Бернес-Ли, Джеймс Хендлер, Ора Лассила. Semantic Web. Scientific American, 2001. (http://ezolin.pisem.net/logic/semantic_web_rus.html)
3. Нариньяни А.С. Кентавр по имени ТЕОН: Тезаурус+Онтология // Труды Международной конференции ДИАЛОГ-2001. – М., 2001. – Т.1. – С.184-188.
4. Росеева О.И., Загорулько Ю.А. Организация эффективного поиска на основе онтологий. Диалог – 2001.
5. OntoWeb www.ontoweb.org
6. Christopher Brewster, Jose Iria, Fabio Ciravegna and Yorick Wilks, The Ontology: Chimaera or Pegasus, presented at the Dagstuhl Seminar Machine Learning for the Semantic Web, 2005
7. Gomez-Perez A., Fernandez-Lopez M., Corcho O. OntoWeb. Technical Roadmap. D.1.1.2. - IST project IST-2000-29243. (www.aifb.uni-karlsruhe.de/WBS/ysu/publications/ OntoWeb_Del_1-1-2.pdf)
8. Gruber T.R., A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993.
9. Guarino,N. Formal Ontology and Information Systems. In N. Guarino, editor, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy, pages 3-- 15. IOS Press, June 1998.
10. Lenat D., Miller G., Yokoi T. CYC, WordNet, and EDR: critiques and responses. - Communications of the ACM. - Volume 38 , Issue 11 (November 1995), pp. 45 - 48.
11. Mahesh K., Nirenburg S., A Situated Ontology for Practical NLP. // Proc. Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), 1995. Montreal, Canada.

2. ТИПЫ ОНТОЛОГИЙ: ВЕРХНЕГО УРОВНЯ, ПРЕДМЕТНЫХ ОБЛАСТЕЙ, ПРИКЛАДНЫХ ОНТОЛОГИИ. ЛЕКСИЧЕСКИЕ ОНТОЛОГИИ.

Классификации онтологий

В проектировании онтологий условно можно выделить два направления, до некоторого времени развивавшихся отдельно. Первое связано с представлением онтологии как формальной системы, основанной на математически точных аксиомах. Второе направление развивалось в рамках компьютерной лингвистики и когнитивной науки. Там онтология понималась, как система абстрактных понятий, существующих только в сознании человека, которая может быть выражена на естественном языке (или какой-то другой системой символов). При этом обычно не делается предположений о точности или непротиворечивости такой системы.

Таким образом, существует два альтернативных подхода к созданию и исследованию онтологий. Первый (*формальный*) основан на логике (предикатов первого порядка, дескриптивной, модальной и т.п.). Второй (*лингвистический*) основан на изучении естественного языка (в частности, семантики) и построении онтологий на больших текстовых массивах, так называемых *корпусах*.

В настоящее время данные подходы тесно взаимодействуют. Идет поиск связей, позволяющих комбинировать соответствующие методы. Поэтому иногда бывает сложно отделить лексические онтологии с элементами формальных аксиоматик от логических систем с включениями лингвистических знаний.

Независимо от различных подходов можно выделить 3 основных принципа классификации онтологий:

- По степени формальности
- По наполнению, содержимому
- По цели создания

Рассмотрим соответствующие классификации по порядку.

Классификация по степени формальности. «Спектр онтологий»

Обычно люди и компьютерные агенты (программы) имеют некоторое представление значений терминов. Программные агенты иногда предоставляют спецификацию входных и выходных данных, которые могут быть использованы как спецификация программы. Сходным образом онтологии могут быть использованы, чтобы предоставить конкретную спецификацию имен терминов и значений терминов. В рамках такого

понимания (где онтология является спецификацией концептуальной модели – *концептуализации*) существует простор для вариаций. Онтологии могут быть представлены как спектр в зависимости от деталей реализации.

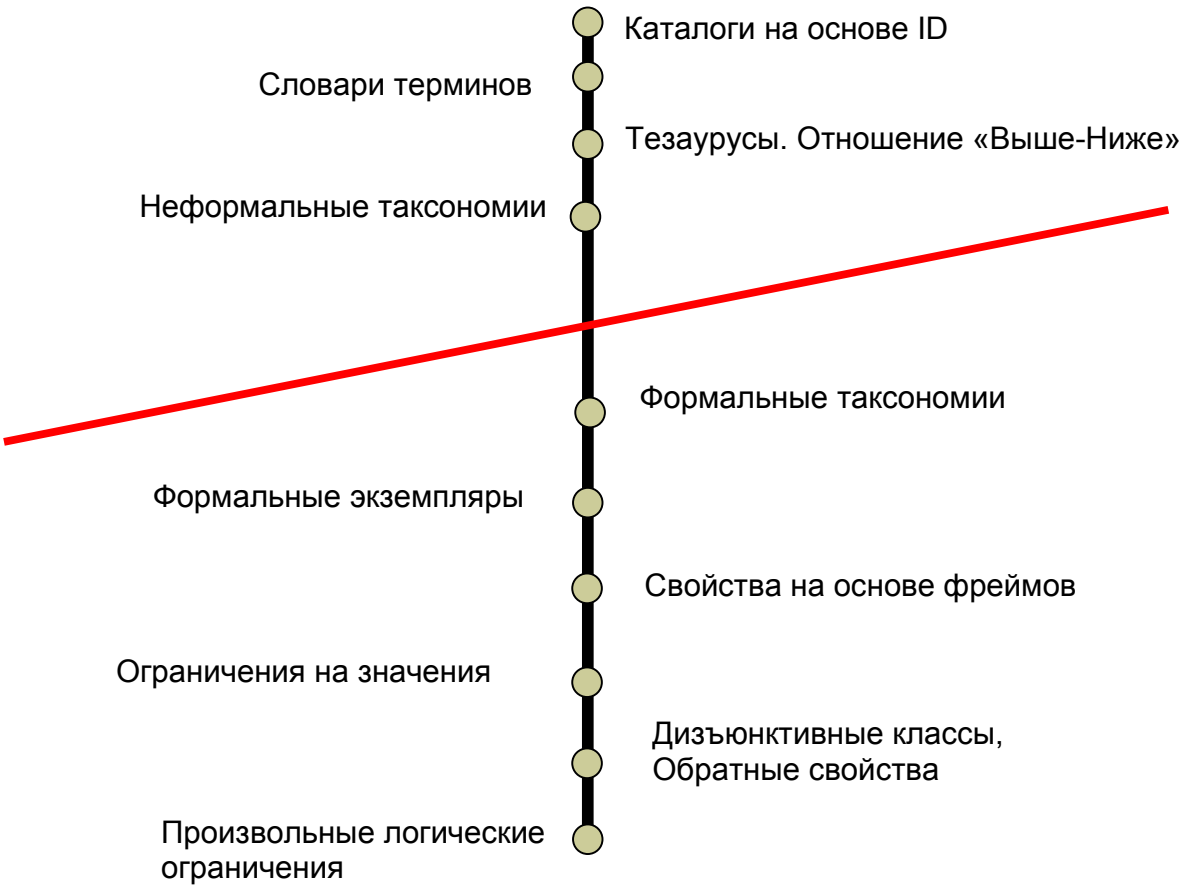

Рис. 1. Спектр онтологий

На рисунке 1 изображен, так называемый, *спектр онтологий* по степени формальности представления, использованию тех или иных формальных элементов. Каждая точка соответствует наличию некоторых ключевых структур в онтологии, отличающих ее от других точек на спектре. Косая черта условно отделяет онтологии от других ресурсов, имеющих онтологический характер.

Первой точке на спектре соответствует контролируемый словарь, т.е. конечный список терминов (простейшим примером является каталог на основе идентификаторов). Каталоги представляют точную (не многозначную) интерпретацию терминов. Например, каждый раз ссылаясь на термин

«машина» мы будем использовать одно и то же значение (соответствующее некоторому ID в словаре), вне зависимости от того, о чем идет речь в контексте: о «стиральной машине», «автомобиле» или «государственной машине».

Другой спецификацией онтологии может быть глоссарий, представляющий список терминов с их значениями. Значения описываются в виде комментариев на естественном языке. Это дает больше информации, поскольку люди могут прочесть такой комментарий и понять смысл термина. Интерпретации терминов могут быть многозначными. Глоссарии непригодны для автоматической обработки программными агентами, но можно, как и ранее, присвоить терминам ID.

Тезаурусы несут дополнительную семантику, определяя связи между терминами. Отношения свойственные для тезаурусов: синонимия, иерархическое отношение и ассоциация. Обычно тезаурусы в явном виде не имеют иерархии терминов, но она может быть восстановлена.

Ранние иерархии терминов, появившиеся в Сети, определяли общие понятия обобщения и уточнения. Yahoo, например, ввела небольшое число категорий верхнего уровня таких как “предметы одежды”. Затем “Платье” определялось как вид (женской) одежды. Явная иерархия, не соответствовала в точности формальным свойствам иерархического отношения (isA). В таких иерархиях может встретиться ситуация, в которой экземпляр класса-потомка также является экземпляром класса предка. Например, общая категория «предметы одежды» включает подкатегорию «женские» (которая должна более точно называться «женские предметы одежды»), а эта категория в свою очередь включает подкатегории «аксессуары» и «платья». Каждый экземпляр категории «платья» является экземпляром категории «предмет одежды» (и, возможно, экземпляром «женского платья» - ведь существуют и «кукольные платья»). Ясно, что экземпляр категории «духи» (как женские «аксессуары») не может быть экземпляром категории «предмет одежды». Здесь не выполняется важное свойство отношения isA – транзитивность.

Далее следует точка «формальные таксономии». Эти онтологии включают точное определение отношения isA (класс-подкласс). В таких системах строго соблюдается транзитивность отношения isA: если B – является подклассом класса A, то каждый подкласс класса B также является подклассом класса A. A для отношения класс-экземпляр (isInstanceOf) выполнятся следующее свойство: если B – является подклассом класса A, то каждый экземпляр класса B также является экземпляром класса A. Поэтому в приведенном выше примере с «духами», «духи» не могут быть помещены ниже в иерархии «предмет одежды» (аксессуары, строго говоря, не являются предметами одежды) или стать экземпляром этой категории. Строгая

иерархия необходима при использования наследования для процедуры логического вывода.

Следующая точка - наличие формального отношения класс-экземпляр. Некоторые классификации включают только имена классов, другие содержат на нижнем уровне экземпляры (*индивиды*). Данная точка спектра включает экземпляры классов.

Далее среди структурных элементов появляются фреймы. Здесь классы (*фреймы*) могут иметь информацию о свойствах (*слотах*). Например, класс «предмет одежды» может иметь свойства «цена», «сделанИз». Свойства бывают особенно полезными, когда они определены на верхних уровнях иерархии и наследуются подклассами. В потребительской иерархии класс «продукт» может иметь свойство «цена», которое получат все его подклассы.

Большей выразительностью обладают онтологии, включающие ограничения на область значений свойств. Значения свойств берутся из некоторого предопределенного множества (целые числа, символы алфавита) или из подмножества концептов онтологии (множество экземпляров данного класса, множество классов). Можно ввести дополнительные ограничения на то, что может *заполнять* свойство. Например, для свойства «сделанИз» класса «предмет одежды» значения можно получать как экземпляры класса «Материал». Легко увидеть, какие проблемы могут возникнуть в этом случае при использовании нестрогой таксономии. Если «духи» - подкласс класса «предмет одежды», то он наследует свойство «сделанИз» вместе с ограничением («Материал»).

В целом с необходимостью выразить больше информации, выразительные средства онтологии (и ее структура) усложняется. Например, может потребоваться заполнить значение какого-либо свойства экземпляра, используя математическое выражение основанное на значениях других свойств и даже других экземплярах. Многие онтологии позволяют объявлять два и более классов дизъюнктивными (непересекающимися). Это означает, что у данных классов не существует общих экземпляров.

Некоторые языки позволяют делать произвольные логические утверждения о концептах – аксиомы.

Языки описания онтология, подобные CycL и Ontolingua позволяют описывать утверждения на языке логики предикатов первого порядка (FOL).

Классификация по цели создания

В рамках этой классификации выделяют 4 уровня (см. рис. 2): Онтологии представления, онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии.

Рис. 2. Классификация онтологий по цели создания.

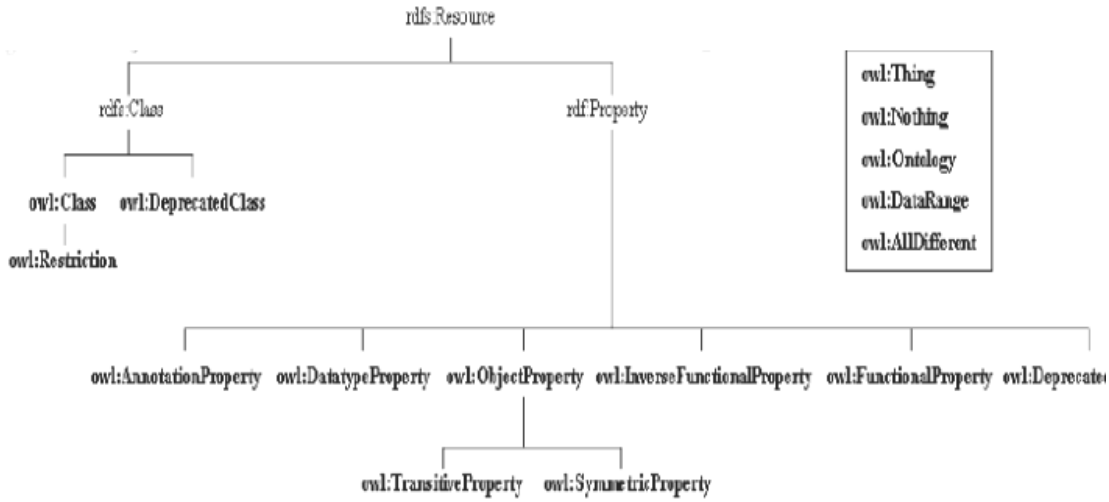

Рис. 3. Онтология представления языка OWL.

Онтологии представления

Цель их создания: описать область представления знаний, создать язык для спецификации других онтологий более низких уровней. Пример: описание понятий языка OWL средствами RDF/RDFS (рис.3).

Онтологии верхнего уровня

Их назначение в создании единой “правильной онтологии”, фиксирующей знания общие для всех предметных областей и многократном использовании данной онтологии. Существует несколько серьезных проектов: SUMO, Sowa’s Ontology, Cyc. Но в целом попытки создать онтологию верхнего уровня на все случаи жизни пока не привели к ожидаемым результатам. Многие онтологии верхнего уровня похожи друг на друга. Они содержат одни и те же концепты: Сущность, Явление, Процесс, Объект, Роль и т.п.

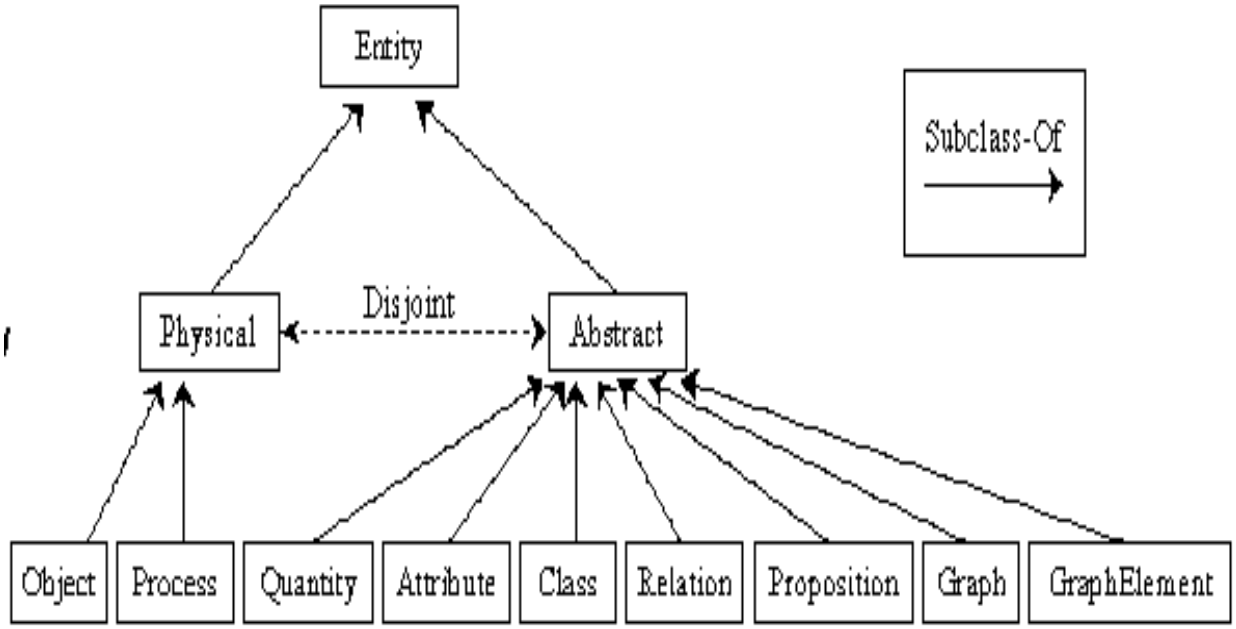

Рис.4. SUMO - онтология верхнего уровня

Онтологии предметных областей

Назначение схоже с назначением онтологий верхнего уровня, но область интереса ограничена предметной областью (авиация, медицина, культура). Примеры: АвиаОнтология, CIDOC CRM, UMLS.

Прикладные онтологии

Назначение этих онтологий в том, чтобы описать концептуальную модель конкретной задачи или приложения. Они содержат наиболее специфичную информацию. Примеры проектов: TOVE, Plinius.

Классификация онтологий по содержимому

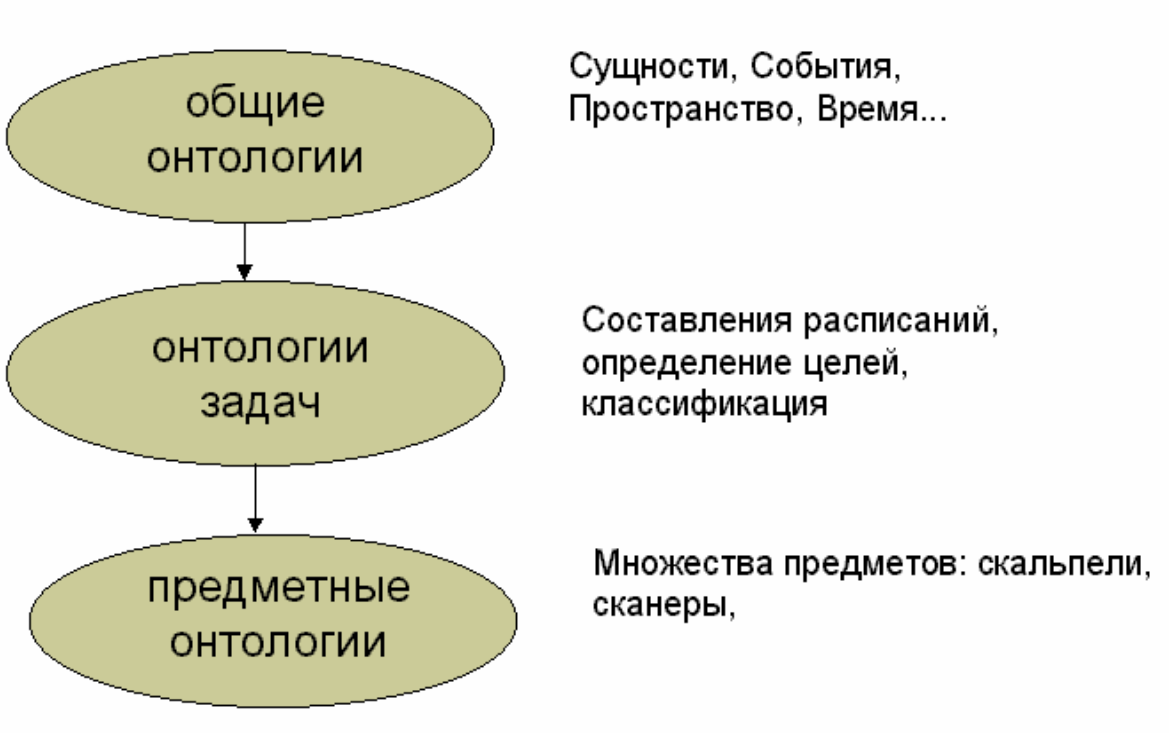

Рис. 5. Классификация онтологий содержимому.

Данная классификация очень похожа на предыдущую, но здесь упор делается на реальное содержимое онтологии, а не на абстрактную цель, преследуемую авторами.

Онтологии и обработка текстов на естественном языке

Для того чтобы применить онтологию для автоматической обработки текстов, в частности для решения задач информационного поиска, необходимо понятиям онтологии сопоставить набор языковых выражений (слов и словосочетаний), которыми понятия могут выражаться в тексте.

Процедура сопоставления понятий онтологий и языковых выражений может быть осуществлена различными способами:

Во-первых, онтология может быть сделана заранее, путем логической классификации, а затем к ее единицам могут быть приписаны языковые единицы. Так, например, Doug Lenat, руководитель известного проекта в области представления знаний CYC, в рамках которого предполагалось формализовать знания здравого смысла (common sense) и использовать их, в

частности, для обработки текстов на естественном языке, считает, что учет значений слов может только запутать ("words are often red herrings"), что значения слов делят мир неоднозначно, а линии деления происходят из самых различных причин: исторических, физиологических и т.п.

Предлагается создавать онтологию путем логического анализа, «сверху-вниз». При этом, имена вводимых понятий (желательно) должны отражать те признаки, которые заложены в основу деления. В результате получаются имена понятий достаточно громоздкие, неестественные, с ними трудно оперировать как разработчикам, так и возможным пользователям.

Другой проблемой такого подхода является то, что при приписывании языковых выражений к логически обоснованной системе понятий получается, что одно и то же слово может соответствовать слишком большому количеству таких «правильных» понятий в зависимости от контекста, возникает излишняя многозначность лексической единицы.

Кроме того, тогда как небольшие онтологии могут быть построены методом сверху-вниз, разработка подробных онтологий для реальных приложений – нетривиальная задача. Более того, во многих предметных областях, знание, нужное для распространения и интеграции, содержится в основном в текстах. Из-за внутренних свойств человеческого языка, непростой задачей является связать знания, содержащиеся в текстах, с онтологиями, даже если бы была построена подробная онтология предметной области.

Некоторые исследователи, такие как известный британский лингвист Йорик Вилкс, считают, что «несмотря на то, что все авторы статей по онтологиям подчеркивают, что понятия являются кирпичиками любой онтологии, мы манипулируем понятиями посредством слов. Во всех онтологиях, которые известны, слова используются, чтобы представлять понятия. Следовательно, то множество явлений в мире, которые не вербализованы, не могут быть смоделированы. Мы можем описать это явление как Онтологическая гипотеза Сепира-Уорфа, то есть то, что не описывается словами, не может быть отражено в онтологии…».

Второе направление, которое обычно обсуждается, это установление соответствий между иерархическими лексическими ресурсами типа WordNet и некоторой онтологией. WordNet-ресурсы описывают лексические отношения между значениями слов, представленные в виде отдельных единиц в иерархической сети – синсетов. Отношения между лексическими единицами в значительной мере отражают отношения объектов внешнего

мира, поэтому такие ресурсы часто рассматриваются как особый вид онтологий – лексические или лингвистические онтологий.

Главной характеристикой лингвистических онтологий является то, что они связаны со значениями (“are bound to the semantics”) языковых выражений (слов, именных групп и т.п.).

Лингвистические онтологии охватывают большинство слов языка, и одновременно имеют онтологическую структуру, проявляющуюся в отношениях между понятиями. Лингвистические онтологии могут поэтому рассматриваться как особый вид лексической базы данных и особый тип онтологии.

Лингвистические онтологии отличаются от формальных онтологии по степени формализации. Поэтому предполагается, что разработчики такого рода ресурсов разрабатывают иерархию лексических значений естественного языка, а для более строгого описания знаний о мире необходимо сопоставить такие ресурсы с какими-либо формальными онтологиями.

Так, содержанием одного из проектов является установление отношений между WordNet и EuroWordNet, с одной стороны, и формальной онтологией SUMO - Standartized Upper Merged Ontology, с другой стороны. Проект состоит в том, чтобы установить соответствие между синсетами WordNet и понятиями онтологии, при котором каждый синсет WordNet либо напрямую сопоставляется с понятием онтологии, либо является гипонимом для некоторого понятия, либо примером понятия онтологии.

Участники другого проекта OntoWordNet считают, что недостаточно провести формальную склейку ресурса типа WordNet и формальной онтологии, необходима значительная реструктуризация исходного лексического ресурса.

Третий путь – попытаться разработать единый ресурс, в котором были бы сбалансированы обе части: система понятий – и система лексических значений, что заключается в разумном разделении этих единиц в создаваемом ресурсе и аккуратном описании их взаимосвязей. Попытка такого подхода реализуется в онтологиях MikroKosmos и OntoSem.

Вопросы к лекции

1. Чем отличаются онтологии верхнего уровня от онтологий предметной области?
2. Чем отличаются онтологии предметной области от прикладных онтологий?
3. Перечислите основные характеристики лексических онтологий.

Литература

1. Lassila O, McGuiness D. The Role of Frame -Based Representation on the Semantic Web. Technical Report. Knowledge Systems Laboratory. Stanford University. KSL-01-02. 2001.
2. Van Heist, G.; Schreiber, T.; Wielinga, B. Using Explicit Ontologies in KBS International Journal of Human-Computer Studies. Vol. 46. (2/3). 183-292. 1997
3. Mizoguchi, R. Vanwelkenhuysen, J.; Ikeda, M. Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. IOS Press. 1995. 46-59.

3. ОНТОЛОГИИ ВЕРХНЕГО УРОВНЯ: ОТЛИЧИТЕЛЬНЫЕ ЧЕРТЫ, РЕШАЕМЫЕ ЗАДАЧИ (ПРИМЕРЫ ПРОЕКТОВ – OPENCYC, SUMO, DOLCE, SOWA'S ONTOLOGY)

Введение

Пренебрегая незначительными различиями в определениях термина «*онтология»,* полученных из разных источников (и приведенных в первой части), будем понимать под онтологией систему, состоящую из множества понятий, их определений и аксиом, необходимых для ограничения интерпретации и использования понятий. При решении прикладных задач онтология часто отождествляется с набором классов (или понятий предметной области), связанных определенным набором отношений (или свойств – бинарных отношений). Базовыми типами отношений являются «ПОДКЛАСС-НАДКЛАСС» (гипонимия), «ЧАСТЬ-ЦЕЛОЕ» (меронимия), "ЭКЗЕМПЛЯР-КЛАСС", "ПРИЧИНА-СЛЕДСТВИЕ", отношение зависимости и др.

Онтологии верхнего уровня описывают, так называемое, *общее знание* о моделируемом мире, формируя общую для онтологий нижних уровней систему понятий. В основном онтологии являются разделяемыми (shared) ресурсами (содержимое онтологии одновременно используется несколькими лицами, группами или сообществами). Разделяемые онтологии (в большей степени это относится к онтологиям верхнего уровня) содержат знания *здравого смысла* (common sense).

Рассмотрим и сравним наиболее масштабные проекты онтологий верхнего уровня.

OpenCyc[1] – открытая для общего пользования часть коммерческого проекта Cyc, на текущий момент наиболее масштабной и детализированной онтологии в области общего знания. База знаний OpenCyc содержит информацию из различных предметных областей: Философия, Математика, Химия, Биология, Психология, Лингвистика и т.д. (http://www.opencyc.com).

Ключевым понятием в проекте OpenCyc является *коллекция.* Любая коллекция может содержать подколлекции и экземпляры. Таким образом, в OpenCyc определены два таксономических отношения: "подколлекция-надколлекция" (genls) и "экземпляр-коллекция" (isa). Экземпляром коллекции может быть любой термин онтологии. Важная черта отношения isa в том, что

[1] OpenCyc – прикладная онтология, в статье рассматриваются только верхние уровни иерархии.

оно передается по иерархии отношения genls, т.е. если A является экземпляром коллекции B и B является подколлекцией коллекции C, то A является также экземпляром коллекции C. В случае, если коллекции A и B связаны отношением genls (A genls B), то это означает, что все экземпляры коллекции A являются также экземплярами коллекции B.

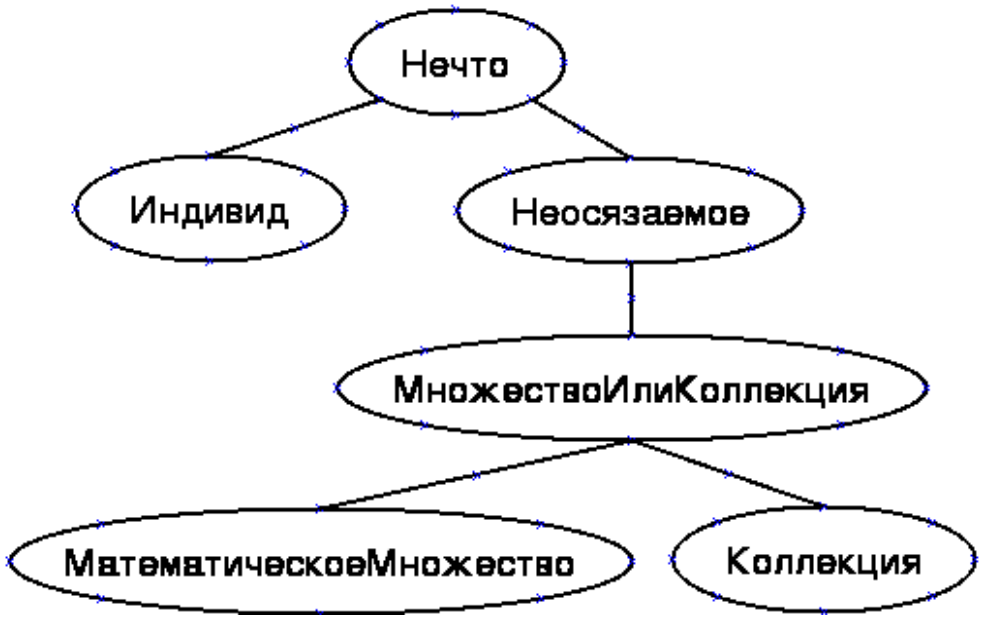

Рис.6. Фрагмент иерархии коллекций в OpenCyc

В вершине иерархии коллекций находится универсальная коллекция с именем "Нечто". По определению, она содержит все, что существует в рамках описываемой области (т.н. «Universe of Discourse»). Любая коллекция описанная в OpenCyc, будь то "Индивид", "МатематическоеМножество" или "Коллекция" является и подколлекцией, и экземпляром коллекции "Нечто". Более того, коллекция "Нечто" является как экземпляром, так и подколлецией самой себя, но не подколлекцией какой-либо другой коллекции. На первом уровне иерархии "Нечто" разделяется сразу на 116 подколлекций. На рис.6 изображена урезанная иерархия коллекций верхних уровней.

Коллекция "Индивид" содержит всевозможные индивиды, т.е. сущности не являющиеся ни множествами, ни коллекциями. Индивиды могут быть абстрактными или конкретными, включать физические объекты, события, отношения, числа, группы, они могут состоять из частей, иметь сложную структуру, но ни один экземпляр этой коллекции не может иметь элементов или подмножеств. Так, индивид имеющий части (связи типа "ЧАСТЬ-ЦЕЛОЕ") и множество или коллекция, содержащая *те же самые части*

(связи типа "ЭЛЕМЕНТ-МНОЖЕСТВО" и "ЭЛЕМЕНТ-КОЛЛЕКЦИЯ") – две совершенно разные сущности. Например, данная фирма (1), группа, содержащая всех работников данной фирмы (2), коллекция всех работников фирмы (3) и множество всех работников фирмы (4) – четыре разных понятия и только первые два из них – индивиды.

Коллекция "Коллекция" содержит все коллекции онтологии OpenCyc, кроме "Нечто". Именно "Коллекция" наиболее близка понятию *класс,* которое часто используют при проектировании онтологий предметных областей (но не понятию *класс* объектно-ориентированного программирования!), поскольку эта коллекция описывает набор объектов (экземпляров коллекции) имеющих некоторые общие атрибуты (свойства). Это же отличает "Коллекцию" от "МатематическогоМножества". Множество может содержать абсолютно не связанные элементы, а “Коллекция” нет. Все экземпляры "Коллекции" являются абстрактными сущностями, даже если коллекция содержит материальные объекты.

Структурно база знаний OpenCyc состоит из констант (терминов) и правил (формул), оперирующих этими константами. Правила делятся на два вида: выводимые утверждения и аксиомы. Под аксиомами в OpenCyc понимаются утверждения, которые были явно и вручную введены в базу знаний экспертами, а не появились там (или могут появиться) в результате работы машины вывода. Все утверждения или формулы в базе знаний OpenCyc фиксируются на языке CycL, выразительно эквивалентном исчислению предикатов первого порядка.

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) – первая из онтологий в библиотеке базовых онтологий проекта WonderWeb. (http://www.loa-cnr.it/DOLCE.html).

Онтологию DOLCE предполагается применять в SemanticWeb для *согласования* между интеллектуальными агентами, использующими разную терминологию. При этом онтология не претендует на звание универсальной, стандартной или общей. Основная цель разработчиков - создать модель, помогающую при сравнении и объяснении связей с другими онтологиями библиотеки WFOL (базовой библиотеки онтологий WonderWeb), а также для выявления скрытых допущений, лежащих в основе существующих онтологий и лингвистических ресурсов, таких как WordNet. DOLCE имеет когнитивный уклон, поскольку фиксирует онтологические категории естественного языка и знания «здравого смысла».

В основу процесса проектирования легло фундаментальное философское разделение всех сущностей на *универсалии* (сущности потенциально или реально имеющие экземпляры) и *индивиды* (или частности), которые не

имеют и не могут иметь экземпляров. DOLCE - онтология индивидов, в том смысле, что область описания ограничена только ими. В качестве примера универсалии можно привести понятие «Собака» (оно имеет множество экземпляров, конкретных примеров в окружающем мире). В отличие от этого понятия, понятие «Время» скорее рассматривается как индивид (едва ли кому-то понадобится трактовать «Время» как множество различных объектов, конечно, если речь не идет о параллельных мирах).

Еще одна черта DOLCE (также заимствованная разработчиками из философии) – явное разделение на «Постоянные» и «Происходящие» сущности. Различие между ними состоит в том, что «Постоянные» сущности имеются в наличии целиком и неизменно в некотором фиксированном промежутке времени (например, стол, дом в течение периода своего существования).

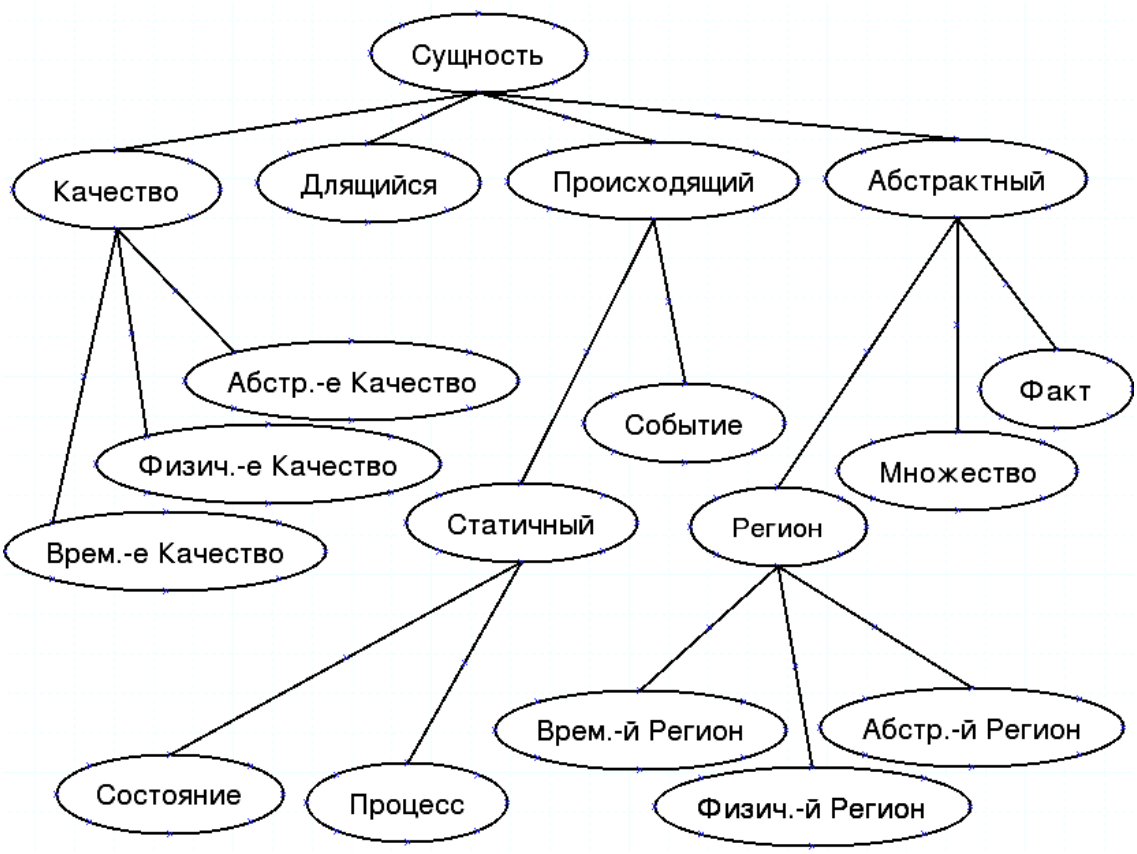

Рис. 7. Верхние уровни иерархии DOLCE.

А «Происходящие» разворачиваются во времени и в каждый момент в некотором временном интервале они могут быть различными, по-разному

себя проявлять, иметь разный состав, (например: ураган, жизненный цикл), однако при этом их идентичность сохраняется.

Такое разделение на "объект" и "процесс" весьма условно и здесь прослеживается когнитивный уклон DOLCE. Оно привело к тому, что в онтологии определены два типа отношения "ЧАСТЬ-ЦЕЛОЕ". Первое никак не зависит от времени, второе имеет временной индекс, определяющий в каких временных рамках, отношение действует. Подобное "раздвоение" наблюдается и для отношения "КАЧЕСТВО-ОБЛАДАТЕЛЬ КАЧЕСТВА". Другие базовые отношения онтологии: "УЧАСТНИК-ПРОЦЕСС", "КОМПОНЕНТ-ЦЕЛОЕ" (компонент входит в состав целого) и отношение зависимости имеют временной индекс. Для сравнения, в онтологии OpenCyc нет явного деления на «Постоянные» и "Происходящие". Поэтому, среди множества отношений в разделе "Части объектов" нет отношения, учитывающего временной аспект: возможное непостоянство данного отношения.

Для представления своей онтологии авторы DOLCE избрали более гибкий, чем в проекте Cyc, подход: онтология фиксируется на бумаге с использованием логики предикатов первого порядка. Затем описывается та часть аксиом, которая может быть представлена на языке OWL. Оставшиеся аксиомы, выраженные на языке KIF[2], добавляются к OWL описаниям в виде комментариев. Таким образом, достигается выразительность уровня KIF[3] и совместимость с OWL. Недостаток такого подхода в том, что приложения, не имеющие информации о действительной структуре OWL документа, не смогут получить доступ к «закомментированным» знаниям.

SUMO (Suggested Upper Merged Ontology) – онтология верхнего уровня, разработанная в рамках проекта рабочей группы IEEE SUO (IEEE Standard Upper Ontology Working Group) и Teknowledge. Проект претендует на статус стандарта для онтологий верхнего уровня (http://ontology.teknowledge.com/).

Онтология **SUMO** содержит наиболее общие и самые абстрактные концепты, имеет исчерпывающую иерархию фундаментальных понятий (около 1 тыс. понятий), а также набор аксиом (примерно 4 тыс.), определяющих эти понятия. Назначение SUMO – содействовать улучшению интероперабельности данных, извлечения и поиска информации, автоматического вывода (доказательства), обработки естественного языка. Онтология охватывает следующие области знания: общие виды процессов и

[2] KIF – аббревиатура от Knowledge Interchange Format.

[3] Диалект OWL DL по выразительности уступает исчислению предикатов первого порядка и, в частности, языку KIF.

объектов, абстракции (теория множеств, атрибуты, отношения), числа и единицы измерения, временные понятия, части и целое, агенты и намерения. SUMO является «канонической» онтологией верхнего уровня: содержит обозримое число концептов и аксиом, имеет ясную иерархию классов, легко расширяется, является итогом объединения различных общедоступных онтологий верхнего уровня (в том числе онтологии Джона Совы (J. Sowa's ontology), о которой речь пойдет ниже). К преимуществам SUMO можно отнести возможность трансляции описания онтологии на любой из основных языков представления знаний, наличие онтологии среднего уровня (MILO), гладко интегрированной с SUMO, несколько дюжин примеров практического применения, а также связь с WordNet – наиболее крупным на настоящий момент тезаурусом, содержащим около 150 тыс. слов повседневного английского языка.

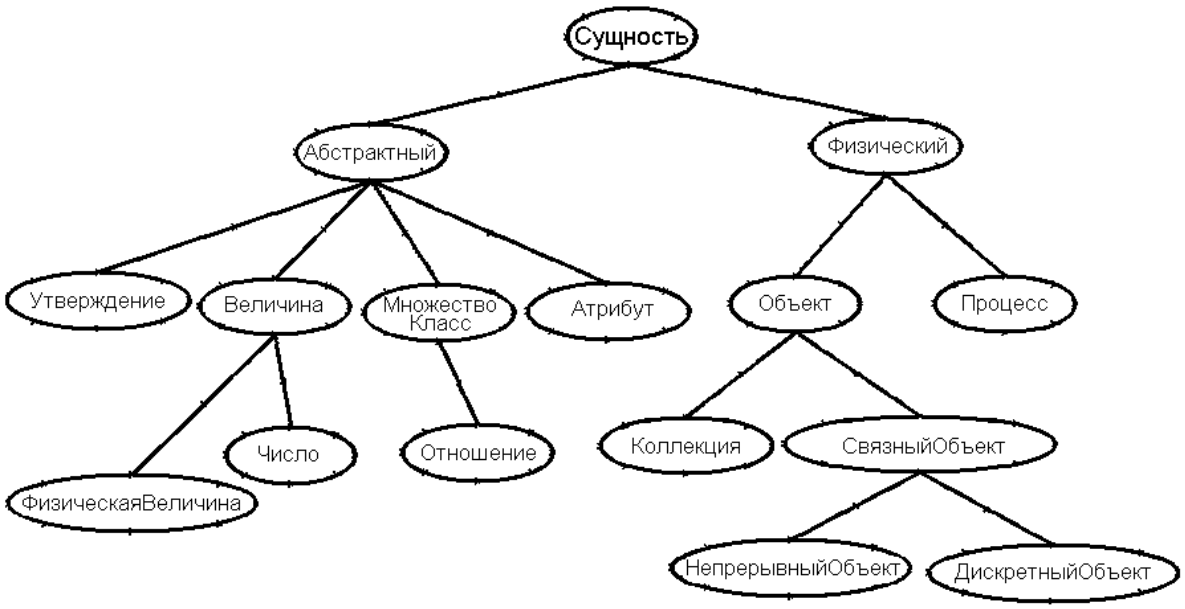

Рис 8. Иерархия классов SUMO.

Иерархия классов в SUMO (рис. 8.) менее запутана, чем в OpenCyc, и, возможно, более удобна для практического применения, чем DOLCE. Основными концептами, как во многих онтологиях верхнего уровня, являются «Сущность» и ее категории – «Физический» и «Абстрактный». Первая категория включают все, что имеет положение в пространстве-времени, а вторая – все остальное (а точнее только то, что существует в воображении). «Физический» делится на «Объект» и «Процесс», что соответствует подходу, реализованному в DOLCE. Непосредственно под концептом «Объект» находятся два непересекающихся понятия:

«СвязныйОбъект» и «Коллекция». Первое обозначает любой объект, все части которого непосредственно или косвенно связаны друг с другом. Концепт «СвязныйОбъект» разделен на два концепта: «НепрерывныйОбъект» и «ДискретныйОбъект» (корпускулярный). «НепрерывныйОбъект» характеризуется тем, что все его части (вплоть до самого низкого уровня деления) имеют такие же свойства, как и целое. Такие субстанции как вода и глина могут быть подклассами концепта «НепрерывныйОбъект», также как и поверхности и географические области. Ниже на диалекте SUO-KIF языка KIF записаны формальные аксиомы, определяющие различие между концептами «НепрерывныйОбъект» и «ДискретныйОбъект».

```
A1. (=>
     (and
          (subclass-of ?OBJECTTYPE НепрерывныйОбъект)
          (instance-of ?OBJECT ?OBJECTTYPE)
          (part-of ?PART ?OBJECT))
     (instance-of ?PART ?OBJECTTYPE))
A2. (equal ДискретныйОбъект (ComplementFn
НепрерывныйОбъект)
```

Аксиома A1 формализует утверждение «Если (PART) является частью объекта (OBJECT), являющегося в свою очередь экземпляром некоторого подкласса (OBJECTTYPE) класса НепрерывныйОбъект, то эта часть (PART) также как и OBJECT является экземпляром класса OBJECTTYPE». Аксиома A2 постулирует факт, что классы ДискретныйОбъект и НепрерывныйОбъект являются взаимодополняющими.

«Коллекции» в SUMO отделены от «СвязныхОбъектов». «Коллекции» состоят из несвязанных частей и отношений («ЧЛЕН-КОЛЛЕКЦИЯ») между частями и соответствующей им коллекцией. Здесь, также как в OpenCyc, проводится разграничение понятий «Коллекция», «Класс» и «Множество». Предикат «быть членом коллекции» отличен от предикатов «быть экземпляром класса» и «быть элементом множества», относящих объекты к «Классам» или «Множествам», которым они принадлежат. В отличие от «Классов» и «Множеств», «Коллекции» занимают некоторое положение в пространстве-времени (они не абстрактны как в OpenCyc, а материальны), члены могут добавляться и удаляться из коллекции, не меняя ее идентичность. Примеры «Коллекций» ящики с инструментами, футбольные команды, отары овец. Возвращаясь к концептам уровня «Физический»-«Абстрактный», обсудим ветвь «Абстрактный». Категория «Абстрактный» разделяется на «Множество», «Утверждение», «Величина» и «Атрибут».

«Множество» - обычное понятие теории множеств, включает «Класс», который в свою очередь имеет подкласс «Отношение». «Класс» понимается как множество со свойством или пересечением свойств, которые определяют принадлежность к «Классу», «Отношение» есть «Класс» упорядоченных пар. «Отношение» по смыслу ближе к «Классу», чем к «Множеству». «Отношение» ограничено только теми упорядоченными парами, которые описывают его содержимое.

Концепт «Утверждение» соответствует понятию семантического или информационного содержимого. Однако SUMO не накладывает никаких ограничений на это содержимое. Это более общее понятие, чем используемое в большинстве онтологий, почти невозможно принципиально разделить абстрактное содержимое, выраженное одним предложением и абстрактное содержимое, выраженное многочисленными речевыми единицами. Примеры «Утверждений»: краткое изложение рассказа, музыкальное содержимое напечатанной партитуры.

Понятие «Атрибут» включает все количества, свойства и т.д., которые не представимы как «Объекты». Например, вместо того, чтобы делить класс «Животные» на «ЖивотныеЖенскогоПола» и «ЖивотныеМужскогоПола», создаются экземпляры «Женщина» и «Мужчина» класса «БиологическийАтрибут», который является подклассом «Атрибута».

Наконец, «Величина» разделяется на «Число» и «ФизическаяВеличина». Первое понимается как независящая от системы измерения величина, а второе как составная величина, состоящая из «Числа» и конкретной единицы измерения.

Аксиомы ограничивают интерпретацию концептов и предоставляют базу для систем автоматизированного рассуждения, которые обрабатывают базы знаний соответствующие онтологии SUMO. Пример аксиомы: «Если **C** является экземпляром процесса *горения*, то существуют *выделение тепла* **H** и *излучение света* **L** такие, что оба они **H** и **L** являются подпроцессами **C**». Более сложные, но логичные предложения говорят, что процессы *выделения тепла* и *излучения света* сопутствуют каждому процессу *горения*. Аксиомы кодируются в SUMO на формальном логическом языке SUO-KIF.

Онтология Джона Совы, предложенная им в книге «Knowledge Representation: Logical, Philosophical, and Computational Foundations», определяет базовые онтологические категории, полученные автором из источников по логике, лингвистике, философии и искусственному интеллекту (http://www.jfsowa.com/ontology/).

Для того чтобы сохранить открытость, онтология, по мнению Совы, должна быть основана не на фиксированной иерархии концептов, а на

каркасе, описывающем различия, по которому иерархия генерируется автоматически. В любом конкретном приложении концепты не определяются рисованием линий на диаграмме, а задаются выбором подходящего множества различий.

	Физический		**Абстрактный**	
	Конти-нуальный	Проис-ходящий	Конти-нуальный	Проис-ходящий
Незави-симый	*Объект*	*Процесс*	*Схема*	*Скрипт*
Относи-тельный	*Слияние*	*Участие*	*Описание*	*История*
Опосред-ованный	*Структура*	*Ситуация*	*Причина*	*Цель*

Рис. 9. Онтологические категории верхнего уровня, предложенные Джоном Совой.

Кроме приведенных на рис.9 категорий в онтологии есть еще два понятия. «Сущность» не определяет никаких отличительных признаков или различий и является надтипом для всех других концептов. Второе понятие – «Абсурдный» тип, наследующий все возможные, в том числе противоречащие, различия. Ни один экземпляр не может иметь этот тип. В онтологии также проводится различие между абстрактным и физическим (именно в таком виде оно заимствовано разработчиками SUMO). Отдельно выделяются категории независимости, относительности и опосредованности. «Независимые» сущности не нуждаются в существовании каких-либо связей с другими сущностями. Любая «Относительная» сущность обязательно имеет х отя бы одну связь с некоторой другой сущностью. Для существования «Опосредованной» сущности необходимо наличие некоторого отношения связывающего какие-то другие сущности, имеющие отношение также и к первой (например, бракосочетание). Онтология Джона Совы описывает роли и отношения, агентов, процессы, и т.д.

WordNet – один из наиболее полно разработанных тезаурусов общего назначения. Здесь мы рассмотрим верхние уровни WordNet как онтологию. Подробное описание структуры WordNet будет изложено в других разделах. Центральным объектом в WordNet является синсет, множество синонимов (или синонимический ряд). WordNet содержит около 70 тыс. синсетов, организованных в иерархию по отношению НАДКЛАСС-ПОДКЛАСС (это отношение также называется гипонимией). Часть иерархии WordNet,

связанная с материальными предметами представлена на рис.10. Здесь можно сразу отметить, насколько верхние уровни WordNet более прозрачны и понятны по сравнению с онтологией Cyc. Для каждого понятия (синсета), есть указатель на существительные, представляющие его части. Например, части для понятия «птица» представляются понятиями «клюв», «крылья» и т.д. Подобные указатели реализуют отношение ЧАСТЬ-ЦЕЛОЕ (меронимия).

В WordNet существуют другие виды связей (например, от существительного к глаголу, чтобы представить функции или к прилагательному, чтобы представить свойства), но не все они реализованы. Эта онтология не имеет аксиом.

В целом WordNet можно представить как сеть, в узлах которой находятся синсеты – лексикализованные понятия. Основными типами отношений являются гипонимия и меронимия.

Из рис.10 видно, что некоторые понятия, «ошибочно» попали на верхние уровни иерархии. Достаточно рассмотреть ряд таксонов (Вещество – Артифакт – Пища(?) – Природный объект) или (Человек(?) – Растение – Животное). По всей видимости, такие «несоответствия» есть результат сильной зависимости структуры онтологии от языка. Понятно, что можно было бы поместить синсет «Человек» под синсет «Животное, животный мир», но, либо, эти синсеты имеют несовместимые подиерархии, либо в языке слова «Человек» и «Животное» имеют сильно отличающиеся значения.

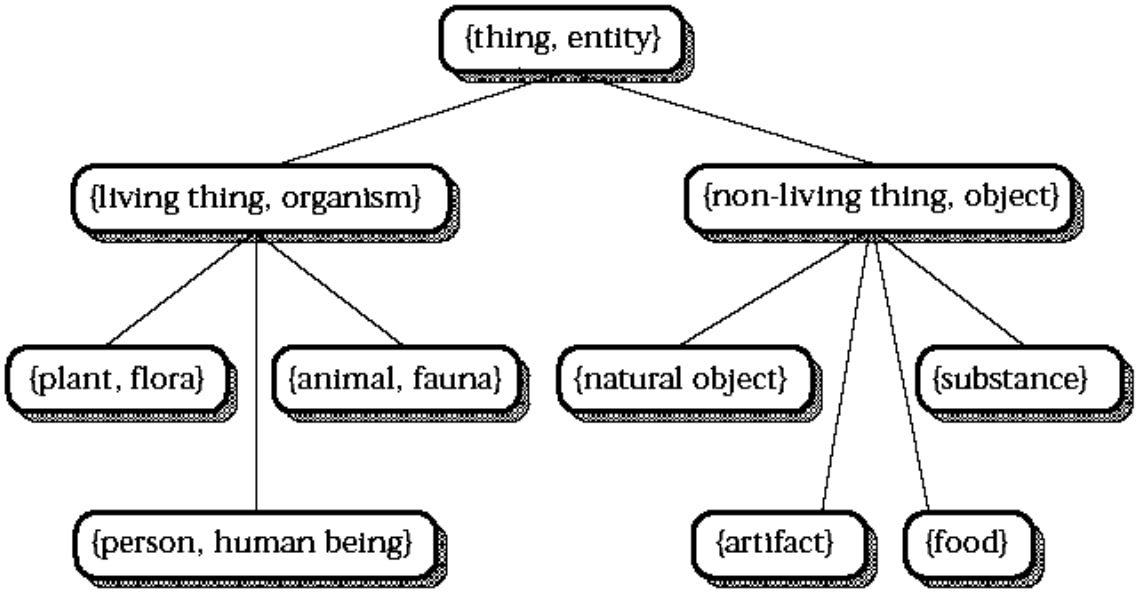

Рис.10. Верхние уровни иерархии синсетов существительных в WordNet.

{act, action, activity}
{animal, fauna}
{artifact}
{attribute, property}
{body, corpus}
{cognition, knowledge}
{communication}
{event, happening}
{feeling, emotion}
{food}
{group, collection}
{location, place}
{motive}
{natural object}
{natural phenomenon}
{person, human being}
{plant, flora}
{possession}
{process}
{quantity, amount}
{relation}
{shape}
{state, condition}
{substance}
{time}

Рис.11. 25 синсетов существительных верхнего уровня WordNet.

Поскольку онтологии верхнего уровня описывают самые общие знания об окружающем мире, они во многом похожи. Так, во всех онтологиях проводится разделение сущностей на абстрактные (такие сущности не могут занимать положения ни в пространстве, ни во времени) и реально существующие (материальные, осязаемые). Во всех онтологиях, так или иначе, присутствует деление на постоянные и временные (меняющиеся во времени) сущности, деление на объект и процесс. В онтологии Джона Совы это деление на «Континуальный» и «Происходящий», в DOLCE – «Постоянные» и «Происходящие», в SUMO – «Объект» и «Процесс».

В то же время даже на верхних уровнях наблюдаются существенные различия. В онтологии SUMO первично разделение на абстрактные и материальные сущности, а разделение на постоянные и временные – вторично. В DOLCE на верхнем уровне производится разделение на постоянные, временные, абстрактные и качественные сущности. В онтологии Совы иерархии сущностей в явном виде нет: в ней описаны только категории, по которым понятия разделяются или группируются. В онтологии OpenCyc на верхнем уровне коллекция "Нечто" делится на «Неосязаемые» и «Индивиды», но экземпляры и тех и других могут быть как абстрактными, так и материальными объектами.

Вопросы к лекции

1. Перечислите известные Вам проекты онтологий верхнего уровня.
2. Что такое *универсалии*?

3. Чем существенно отличается отношение НАДКЛАСС-ПОДКЛАСС от ЧАСТЬ-ЦЕЛОЕ?

Литература

1. Lassila O, McGuiness D. The Role of Frame -Based Representation on the Semantic Web. Technical Report. Knowledge Systems Laboratory. Stanford University. KSL-01-02. 2001.
2. Van Heist, G.; Schreiber, T.; Wielinga, B. Using Explicit Ontologies in KBS International Journal of Human-Computer Studies. Vol. 46. (2/3). 183-292. 1997
3. Mizoguchi, R. Vanwelkenhuysen, J.; Ikeda, M. Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. IOS Press. 1995. 46-59.

4. НАЗНАЧЕНИЕ ОНТОЛОГИЙ. ЗАДАЧИ РЕШАЕМЫЕ С ПОМОЩЬЮ ОНТОЛОГИЙ И ТЕЗАУРУСОВ (ИНФОРМАЦИОННЫЙ ПОИСК, ИНТЕГРАЦИЯ ГЕТЕРОГЕННЫХ ИСТОЧНИКОВ ДАННЫХ, SEMANTIC WEB)

Введение

Как было отмечено, существует множество способов классифицировать онтологии. Предполагаемая область применения и цель могут влиять на масштаб и содержимое онтологии. Существенным является ответ на вопрос, какую пользу может принести использование онтологий при решении то или иной задачи.

В данном разделе описываются проблемные области, в которых можно было бы достичь дополнительных преимуществ при использовании онтологий:

- в вычислительном плане (например, для сокращения времени вычислений);
- в экономическом плане (например, для сокращения затрат на разработку программного обеспечения, интеграцию данных);

по сравнению с уже существующими решениями (основанными на классических подходах).

Но для доказательства преимуществ онтологического подхода необходимо проводить сравнение реально действующих проектов, ставить эксперименты. Такая задача не под силу какой-либо одной организации. Важно привлечение ресурсов многих предприятий в рамках одной инициативы. Наиболее крупной «площадкой» для экспериментов с семантическими технологиями в настоящее время является проект Semantic Web. Хотя за время прошедшее со старта проекта в 2001 году, в концептуальном виде структура Сети (World Wide Web) не изменилась, появился ряд критически важных средств, технологий и языков описания онтологий, необходимых для дальнейшего развития Semantic Web.

Второй по масштабу задач можно условно считать область информационного поиска (Information Retrieval).

В современных поисковых системах тексты автоматически индексируются по набору составляющих эти тексты слов.

Такое представление текстов как простого наборы слов (“bag of words”) имеет большое количество очевидных недостатков, затрудняющих поиск релевантных текстов, таких как

- избыточность -- в пословном индексе используются слова-синонимы, выражающие одни и те же понятия;
- слова текста считаются независимыми друг от друга, что не соответствует свойствам связного текста;
- многозначность слов --поскольку многозначные слова могут рассматриваться как дизъюнкция двух или более понятий, выражающих различные значения многозначного слова, то маловероятно что все элементы этой дизъюнкции интересуют пользователя.

Этих недостатков лишено так называемое концептуальное индексирование, то есть такое индексирование, когда текст индексируется не по словам, а по понятиям, которые обсуждаются в данном тексте. При такой технологии

- все синонимы сведены к одному и тому же понятию,
- многозначные слова отнесены к разным понятиям
- связи между понятиями и соответствующими словами описаны и могут быть использованы при анализе текста.

Для того, чтобы попытаться реализовать схему автоматического концептуального индексирования и концептуального поиска необходимо иметь ресурс, описывающий систему понятий данной предметной области, то есть онтологию в данной предметной области.

Нужно отметить, что использование онтологий для информационного поиска в реальных широких предметных областях имеет ряд особенностей:

- эта онтология должна быть очень большой величины,
- понятия онтологии должны иметь аккуратно установленные связи с языковыми единицами – терминами предметной области,
- онтология реальной предметной области не может быть полной, поэтому методы информационного поиска на основе онтологий должны сочетаться с методами информационного поиска на основе пословных методов в едином поисковом механизме,
- задача информационного поиска предполагает использование онтологий для анализа свободных неограниченных связных текстов, для которых не существует хорошо развитых методов автоматической обработки.

Все эти факторы ограничивают внедрение онтологий в поисковые механизмы информационно-поисковых систем. Сложившаяся ситуация означает, что необходимы новые исследования:

- для выяснения, онтологии какого типа могут быть наиболее эффективными при их использовании в задачах информационного поиска;
- для разработки комбинированных технологий информационного поиска, сочетающих методы на базе знаний, описанных в онтологиях, так и пословные методы,
- для разработки методов «быстрого» создания онтологий для широких предметных областей на основе текстовых коллекций и интеграции знаний уже существующих онтологий.

Подобные трудности характерны и в области интеграции разнородных баз данных. Но здесь преимущества от внедрения решений на базе онтологий могут быть более ощутимыми для конкретного предприятия. Поскольку система понятий ограничена предметной областью и объемы данных не настолько впечатляющи как в Internet, построение соответствующей онтологии вполне возможно, т.к. каждому приложению (или БД) не важно какую именно онтологию верхнего уровня оно разделяет с другими. Важно, использование *единой* онтологии верхнего уровня или даже онтологии предметной области, которая может способствовать интеграции данных, обеспечить интероперабельность. Но эти размышления не имеют под собой научной базы, они не проверены на практике. Подтвердить или опровергнуть их возможно, только применяя онтологии на практике.

В трех следующих подразделах описываются наиболее вероятные области применения онтологий и возможные варианты использования, в которых онтологии могут оказаться наиболее полезными.

Semantic Web

Идея Семантической Сети (Semantic Web) впервые была провозглашена в 2001 Тимом Бернерсом-Ли (создателем World Wide Web). Однако она не является новой ни для автора, ни для web-сообщества в целом. Суть ее состоит в автоматизации «интеллектуальных» задач обработки *значения* (в семантическом смысле) тех или иных ресурсов, имеющихся в Сети. Обработкой и обменом информации должны заниматься не люди, а специальные интеллектуальные агенты (программы, размещенные в Сети). Но для того, чтобы взаимодействовать между собой агенты, должны иметь общее (разделяемое всеми) формальное представление значения для любого

ресурса. Именно для цели представления общей, явной и формальной спецификации значения в Semantic Web используются онтологии.

За пять лет прошедших с первой публикации о Semantic Web, был разработан целый ряд стандартов и рекомендаций, реализовано множество проектов. Но, несмотря на отдельные успехи, до сих пор (и это признает сам Т. Бернерс-Ли) нельзя сказать, что идея Semantic Web реализована на практике. В этом разделе будут изложены предпосылки к созданию Semantic Web, путь который был проделан исследователями с 2001 по 2006 годы и препятствия, возникшие на этом пути.

Работа над средствами описания семантики в Сети началась задолго до публикации 2001 года. В 1997 консорциум W3C определил спецификацию RDF (Resource Description Framework). RDF предоставляет простой, но мощной язык описания ресурсов, основанный на триплетах (*triple-based*) «Субъект-Предикат-Объект» и спецификации URI. В 1999 RDF получает статус рекомендации. Этот шаг в направлении улучшения функциональности и обеспечения интероперабельности (т.е. возможности обмениваться данными, несмотря на их разнородность) в Сети считается одним из важнейших. Концептуально RDF дает минимальный уровень для представления знаний в Сети. Спецификация RDF опирается на ранние стандарты, лежащие в основе Web:

- Unicode служит для представления символов алфавитов различных языков,
- URI используется для определения уникальных идентификаторов ресурсов,
- XML и XML Schema – для структурирования и обмена информацией и для хранения RDF (XML синтаксис RDF).

Кроме RDF был разработан язык описания структурированных словарей для RDF – RDF Schema (RDFS). Он предоставляет минимальный набор средств для спецификации онтологий. RDFS получил статус рекомендации W3C в 2004 году. Однако препятствием для Semantic Web стало то, что документов написанных на языке RDF/RDFS было относительно мало. В период с 2001 по 2004 годы шла интенсивная работа по созданию программных средств для обработки и автоматической генерации RDF-документов.

Результатом в 2004 году стал язык GRDDL (Gleaning Resource Descriptions form Dialects of Languages). Его назначение состоит в предоставлении средств для извлечения RDF-триплетов из XML и XHTML данных (в особенности это относится к документам, автоматически генерируемым из закрытых баз данных). Развивалось и программное

обеспечение для Semantic Web. В области создания библиотек классов и построения логических выводов над RDF-графами была создана библиотека Jena Framework. В области создания модулей расширения для браузеров – Simile для Firefox. В области создания визуальных сред редактирования – большое число редакторов онтологий, стали поддерживать RDF.

В 2004 году статус рекомендации получил язык OWL (Web Ontology Language). Он имеет 3 диалекта (3 множества структурных единиц), используемых в зависимости от требуемой выразительной мощности. OWL фактически является надстройкой над RDF/RDFS и поддерживает эффективное представление онтологий в терминах классов и свойств, обеспечение простых логических проверок целостности онтологии, связывание онтологий друг с другом (импорт внешних определений). Многие формализмы описания знаний могут быть отображены на формализм OWL (один из его диалектов OWL DL основан на упоминавшейся выше дескриптивной логике). Большое число создаваемых в настоящее время онтологий кодируются на OWL, уже существующие онтологии транслируются в него.

На этом работа по обеспечению Semantic Web необходимыми стандартами не остановилась. В 2005 году началась работа над форматом обмена правилами – RIF (Rule Interchange Format). Его назначение – соединить в одном стандарте несколько формализмов для описания правил (по которым может осуществляться нетривиальный логический вывод): логику клауз Хорна, логики высших порядков, продукционные модели и т.п.

В настоящее время проходит последнюю стадию перед рекомендацией язык SPARQL. Это – язык запросов к RDF-хранилищам. Синтаксически он очень похож на SQL.

На рис.11 представлена диаграмма, называемая иногда стеком (или даже «слоеным пирогом») Semantic Web.

Все основные уровни диаграммы были описаны выше. Уровням «Ontology vocabulary» и «Logic» соответствуют OWL и RIF. Уровень «Trust» на данный момент остается незатронутым никакими стандартами. Здесь и возникает одно из существенных препятствий к реализации всей идеи: поддержка автоматической проверки корректности и *правдивости* информации. В самом деле, у многих поставщиков семантических описаний может возникнуть соблазн «обмануть» программу-агента, предоставив информацию не соответствующую действительности, либо навязчивую рекламу, как это в настоящее время проделывается с поисковыми машинами, спам-фильтрами и т.п.

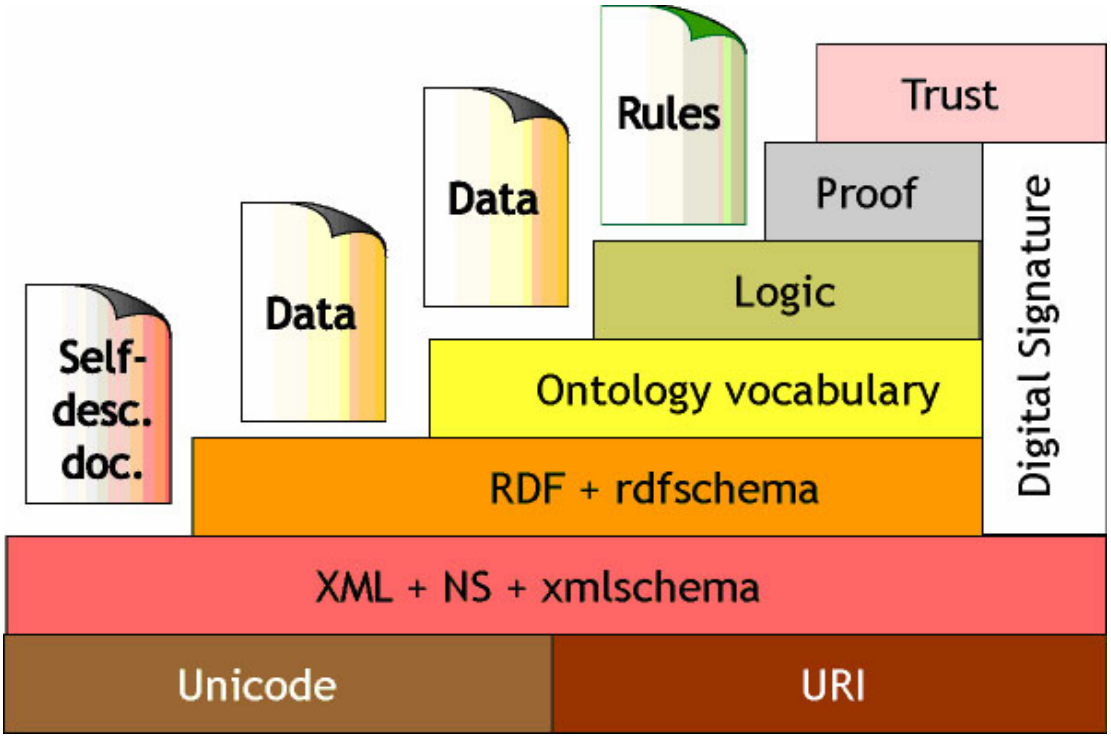

Рис. 11. Диаграмма Semantic Web.

Еще одним камнем преткновения для создания Semantic Web является фактическое отсутствие работающих интеллектуальных агентов. Не всякая программа обрабатывающая RDF, является агентом Semantic Web, точно так же как и не всякая программа, написанная на ПРОЛОГЕ, является приложением в области искусственного интеллекта.

Завершая раздел, нужно подчеркнуть, что Семантическая Сеть продолжает развиваться: появляются новые стандарты.

Один из таких «свежих» стандартов – SPARQL пока еще (июль 2006) не получил статус рекомендации W3C, но уже широко используется разработчиками информационных систем. Новый шаг – начало разработки формата обмена правилами, построенными над онтологиями: RIF (Rule Interchange Format), и определение требований и области его применения. Появилось множество свободно распространяемых библиотек для разработки приложений «под Semantic Web». Главными задачами, стоящими перед сообществом Семантической Сети, остаются: создание новых онтологий и согласование существующих.

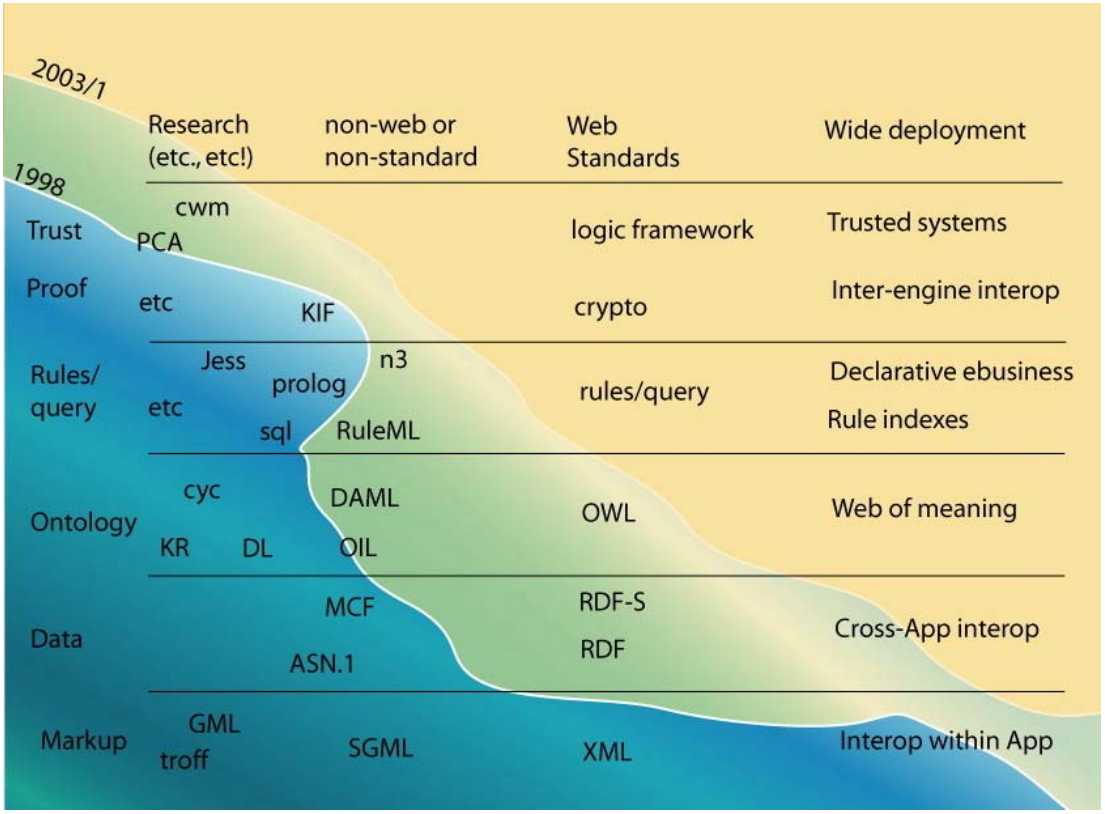

Рис. 12. «Приливная волна» Semantic Web.

На рис. 12 наглядно видна тенденция последних 3-7 лет, которую можно условно назвать «прилив Семантической Сети».

Информационный поиск

Определим базовые понятия в области извлечения документов:

Коллекция – множество документов, имеющих какие-либо общие свойства (например, коллекция документов по заданной тематике, или коллекция документов имеющих только общий формат представления).

Документ – минимальная структурная единица информации, с точки зрения хранения и извлечения из коллекции. Текстовый документ может быть представлен последовательностью более мелких единиц: абзацев, предложений, слов, которые в определенном контексте также являются документами.

При упоминании области информационного поиска – Information Retrieval (IR) обычно имеют в виду комплексную деятельность по сбору, организации, поиску, извлечению и распространению информации при помощи компьютерных технологий. Теоретическими и инженерными

аспектами реализации этих технологий занимаются соответствующие научные и инженерные дисциплины.

Примерами задач в области информационного поиска являются:

- собственно информационный поиск документов по запросу пользователя
- автоматическая рубрикация документов по заранее заданному рубрикатору,
- автоматическая кластеризация документов – разбиение на кластеры близких по смыслу документов,
- разработка вопросно-ответных систем – поиск точного ответа на вопрос пользователя, а не целого документа,
- автоматическое составление аннотации документа и многие другие.

Как любая деятельность, IR имеет цель – обеспечить удовлетворение потребности в информации (или информационной потребности) пользователя. Но сама информационная потребность представляет собой некое психологическое состояние человека. Поскольку в настоящее время не существует носителей для такого рода информации, то единственный способ выразить информационную потребность состоит в ее изложении в форме, доступной для обработки машиной. Например, в форме запроса написанного на естественном языке (ЕЯ). Такой способ очень удобен для человека, но может повлечь несколько неприятностей. Одна из неприятностей состоит в том, что запрос может иметь несколько разных толкований (трактовок). Он может быть неполным, или избыточным, содержать многозначные слова, сильно зависеть от контекста, плохо отражать информационную потребность и т.п. При этом информационная потребность в каждый момент времени остается постоянной, она не допускает альтернативных или взаимоисключающих трактовок. Хотя, запрос редко в точности соответствует информационной потребности, это единственный способ связи между пользователем и поисковой машиной.

Поисковая машина (*поисковик*) использует запрос как входные данные для получения того или иного результата – выборки из коллекции документов, соответствующих запросу (поисковая машина находит документы *релевантные* запросу). Здесь возникает вторая неприятность: пользователь оценивает результат поиска в соответствии со своей информационной потребностью, а не в соответствии с введенным запросом. В ходе оценки он принимает решение о *релевантности* (или мере соответствия) результата поиска и его (пользователя) информационной потребности. Такую оценку может произвести только сам пользователь. Соответствующие суждение о релевантности и саму релевантность называют

истинными. Релевантность, вычисляемая поисковиком на основе его внутренней логики, может не соответствовать истинной релевантности.

Например, пользователя написавшего запрос «школы бальных танцев России» могут интересовать как различные школы бальных танцев, расположенные на территории России, так и школы (в смысле «школа Нуриева»). Пользователь может иметь цель поступить в эту школу, узнать какие педагоги ведут занятия или просто найти партнера по танцам. Все эти намерения скрыты от поисковой машины и не могут быть использованы для вычисления релевантности.

Существуют и другие способы определения релевантности. Например, с учетом полезности: *тематическая релевантность* и *утилитарная релевантность*. Данный документ может точно соответствовать информационной потребности пользователя по теме (*тематическая*), но при этом быть совершенно бесполезным для выполнения конкретной деятельности, в рамках которой возникла потребность (*утилитарная*).

С точки зрения пользователя работа поисковой машины начинается после отправки запроса. Фактически, этому предшествует важный этап индексирования коллекции документов. Он заключается в создании индексных таблиц, значительно ускоряющих обработку запросов. Идея индексирования массивов данных, для ускорения доступа применяется повсеместно. Примером индекса может служить алфавитный или предметный указатель в конце книги, оглавление. Даже закладки сделанные в определенных местах книги являются своего рода индексом. Индексы широко применяются для ускорения доступа в СУБД. Особенность индексирования в IR состоит в том, что индекс необходимый для полнотекстового поиска в электронных коллекциях является наиболее полным. Он должен содержать все термины, которые появляются в документах коллекции. Индекс, содержащий все термины появляющиеся в документах коллекции, называется обратным (или инвертированным) файлом. Часто вместо всевозможных форм каждого слова в инвертированный файл включают только *токены* – части слов, остающиеся после отсечения окончаний. Например, словам «столы», «столу», «столом» соответствует единственный токен «стол». К числу токенов также относят числа, буквенно-числовые коды, аббревиатуры и т.п.

Для каждого токена на множестве документов вычисляются следующие характеристики:

1. число документов, в которых появился данный токен (эта характеристика говорит о распространенности токена в коллекции);

2. частота встречаемости токена в коллекции (эта характеристика показывает насколько данный токен «необычен» по сравнению с другими).

Кроме того, для данной пары «токен – документ (содержащий данный токен)» в инвертированном словаре хранится информация о:

1. частоте встречаемости токена в документе;
2. смещении токена от начала документа. Оно может измеряться в символах, или в словах. Эта характеристика часто используется для быстрого извлечения слов контекста и для составления краткого резюме.

Для многих подходов к организации полнотекстового поиска оказывается достаточно инвертированного файла или подобной структуры данных. Но при извлечении релевантных данному запросу документов необходимо определить способ обработки запроса и вычисления значения релевантности для каждой пары «запрос-документ».

Способы обработки запроса

Исторически одним из первых способов обработки запросов был, так называемый, ***«Булев поиск»***. В этом подходе слова запроса соединяются между собой логическими связками: AND ($\&$), OR($\vee$), NOT($\neg$). Допустима группировка при помощи скобок. Таким образом, запрос представляется логической формулой, в которой атомами могут быть термины или какие-либо дополнительные условия (ограничение на число любых слов между двумя заданными словами, поиск только в том же параграфе или предложении текста, поиск точной фразы и т.п.). Поисковая машина, основанная на булевом поиске, возвращает документы, для которых формула-запрос принимает истинные значения. Каждому атому формулы сопоставляется множество документов, для которых значение атома истинно. Если атом является термином, то ему сопоставляется множество документов, в которых термин встречается. Затем над множествами выполняются элементарные операции: объединения, пересечения и дополнения, соответствующие логическим связкам между атомами:

$$T_1 \vee T_2 \sim D_{T_1} \bigcup D_{T_2},$$

$$T_1 \,\&\, T_2 \sim D_{T_1} \bigcap D_{T_2} ,$$

$$\neg T \sim D_0 \setminus D_T ,$$

где T , T_1 и T_2 - атомы, D_T - множество документов, для которых атом T принимает истинное значение, D_0 - множество всех документов коллекции.

Такой подход к обработке запроса имеет ряд недостатков.

1. На данный запрос поисковая машина может вернуть очень много документов (или даже все документы коллекции). В этом случае пользователь вынужден последовательно добавлять условия в запрос, чтобы уменьшить результирующую выборку. Поиск производится методом «проб и ошибок».
2. Как правило, полезную выборку обозримого размера можно получить, задав сложную логическую формулу. При этом от пользователя требуется не только знания правил построения формул, но и достаточно хорошее знакомство с «языком» предметной области.
3. Вследствие того, что существует только два значения релевантности: «релевантен» (true) и «нерелевантен» (false), результирующая выборка не может быть упорядочена по релевантности. Все документы одинаково релевантны.
4. Все атомы формулы имеют одинаковую важность (вес), хотя, некоторые из них могут быть «ключевыми», другие – вспомогательными.

Существуют способы улучшения качества булевого поиска. Для автоматического расширения запроса синонимичными терминами можно использовать тезаурус или другой ресурс онтологического характера.

Негативные стороны булевого поиска связаны с формализмом обработки запроса. Для их устранения необходимо изменить сам подход. Однако, тот факт, что данный подход имеет недостатки, не означает, что от него нужно полностью отказаться. Многие поисковые системы используют булев поиск как альтернативу (обычно под заголовком «Расширенный поиск», что указывает на необходимость дополнительных знаний и навыков пользователя).

Основным способом обработки запросов поисковыми машинами в Интернет является ***ранжированный поиск***. Он основан на вычислении релевантности через распределение частот встречаемости терминов запроса по документам коллекции. На вход может поступать запрос на естественном языке. В процессе предобработки из запроса удаляются *стоп-слова* (например, «где», «почему» и т.п.), частицы. Термины сокращаются до токенов. После этого на основе токенов можно было бы автоматически сформировать логическую формулу. Но эксперименты показали, что связывание атомов операцией AND дает слишком мало документов в результирующей выборке, и многие релевантные документы остаются за ее пределами (ситуация «выплескивания ребенка вместе с водой»). Связывание атомов формулы операцией OR дает противоположный результат: выборка сильно зашумляется. В данном случае булев подход к обработке естественно-языкового запроса не является адекватным. В этой ситуации дополнительная информация о взаимосвязях терминов (онтология) могла быть использована для формирования более сложной логической формулы.

Вместо того, чтобы представлять документы как множества слов, а запрос как последовательность операций над множествами (как в булевом поиске), предлагается следующее.

Каждый документ коллекции представляется вектором в векторном пространстве, размерность которого равна числу токенов в инвертированном файле. Документ описывается «весами» (координатами) соответствующих токенов. Координатные оси являются попарно ортогональными (токены попарно независимы и образуют базис пространства). Запрос, прошедший предобработку, содержит последовательность токенов и также как любой документ может быть разложен по базису пространства. Далее для вычисления релевантности определяется функция, которая каждой паре векторов ставит в соответствие число из отрезка [0,1]. Крайние точки соответствуют значениям «нерелевантен» и «полностью релевантен». Промежуточные значения определяют степень релевантности документа запросу (или двух документов коллекции, если требуется найти «похожие» документы). Рассмотрим подробнее, как определяются значения весов документов и функция релевантности.

Пусть $D = (d_1,..,d_N)$ - множество документов коллекции, $T = (t_1,..,t_M)$ - множество токенов. Для каждого фиксированного *i,* документ d_i представляется вектором весов

$$w_{ji} = tf_{ji} \cdot idf_{ji}, \; j = 1..M \;,$$

где tf_{ji} - частота встречаемости токена t_j в документе d_i по сравнению с другими токенами документа, idf_{ji} - величина обратная частоте встречаемости токена t_j по всем документам коллекции.

Фактически, вместо idf_{ji} используется $\log\left(\frac{N}{n_j}\right)$, где N – число документов в коллекции, n_j - число документов, в которых встретился t_j.

Таким образом, документы коллекции представляются векторами с M координатами. Запрос тоже может рассматриваться как документ и представляется вектором $q=(q_1,..,q_M)$. Матрица W – составленная из векторов документов w_{ji} имеет размерность $M \times N$. Умножая вектор q на W слева, получим вектор $a=(a_1,..,a_N)$, содержащий значения близости между запросом и всеми документами коллекции. После нормирования $a_k, k=1..N$ на модуль вектора q и вектора $w_k=(w_{1k},...,w_{Mk})$, вектор a будет содержать значения релевантности для каждой пары (q,d_k). Иными словами, документы коллекции ранжируются по релевантности данному запросу, что полезно при представлении результатов поиска.

Еще одним подходом к обработке запроса является ***вероятностная модель***. Это попытка описать ранжированный поиск в терминах теории вероятностей. Проблема состоит в том, что частоты, используемые в ранжированном поиске по своему смыслу, не имеют никакого отношения к вероятностям. Число появлений термина в документе не может служить значением случайной величины и использоваться для оценки вероятности появления данного термина в других документах коллекции. Поэтому частоты встречаемости терминов нельзя использовать в стандартных формулах теории вероятностей.

В основу модели положен способ вычисления вероятности того, что данный документ релевантен запросу. В случае если вероятность достаточно велика, документ считается релевантным.

Основные предположения заключаются в следующем:

1. Документ d либо релевантен, либо нерелевантен запросу q (т.е. для каждого события (d,q) возможно только 2 элементарных исхода (w_0,w_1)).

2. Определение одного документа как релевантного не дает никакой информации о релевантности других документов.

Таким образом, теория не учитывает ни степень релевантности, ни то, что релевантность одного документа может влиять на релевантность других. Этот способ вычисления релевантности далек и от определения *истинной релевантности,* и от определения *полезной релевантности*. Однако сами значения вероятности релевантности могут быть полезны при представлении результатов (для упорядочивания выборки).

Вероятность извлечения из коллекции документа D релевантного запросу Q может быть выражена так:

$$P\left(R_Q = X \mid D\right), \qquad (*)$$

где R_Q - случайная величина, принимающая значения из множества $X = \{0,1\}$. $R_Q = 1$ соответствует событию извлечения из коллекции релевантного документа, $R_Q = 0$ – нерелевантного.

Выражение (*) для того, чтобы его можно было вычислить оценивается так:

$$\frac{P\left(R_Q = 1 \mid D\right)}{P\left(R_Q = 0 \mid D\right)} = \frac{P\left(R_Q = 1\right)P\left(D \mid R_Q = 1\right)}{P\left(R_Q = 0\right)P\left(D \mid R_Q = 0\right)} \approx \frac{P\left(D \mid R_Q = 1\right)}{P\left(D \mid R_Q = 0\right)}.$$

Отношение в левой части равенства («близость» документа и запроса) показывает насколько вероятность извлечь из коллекции релевантный запросу документ больше вероятности извлечь нерелевантный документ.

Отношение после знака эквивалентности оценивается с использованием распределения терминов запроса по документам коллекции:

$$\frac{P\left(D \mid R_Q = 1\right)}{P\left(D \mid R_Q = 0\right)} \approx \prod_{t \in Q} \frac{P(t \mid R_Q = 1)}{P(t \mid R_Q = 0)}.$$

Эксперименты показали, что качество работы поисковых машин на основе вероятностной модели в целом не лучше поисковиков, основанных на ранжированном поиске.

Оценка поисковых машин

С ростом числа поисковых машин, различных методик, алгоритмов поиска, возникла необходимость сравнивать качество их работы. Для этого были введены две характеристики: *точность (p)* поиска и его *полнота (r)*.

Полнота (recall, r) – доля релевантных документов в выборке, по отношению ко всем релевантным документам коллекции.

Точность (precision, p) – доля релевантных документов выборки, по отношению ко всем документам в выборке.

Эти два критерия обычно конфликтуют. Стопроцентная точность и полнота на практике недостижимы.

Пусть N – число документов в коллекции, n – число документов в коллекции релевантных некоторому запросу, m – число документов в выборке полученной системой на данном запросе, A – число релевантных документов в выборке. Тогда

$$p=A/m, \quad r=A/n,$$

	Релевантные	**Нерелевантные**	
Извлечены	A	B	A + B = m
Не извлечены	C	D	C + D = N – m
	A + C = n	B + D = N - n	A + B + C + D = N

Попытки улучшить качество информационного поиска на основе онтологических ресурсов

Конкретные примеры использования онтологий для приложений информационного поиска в рамках булевской и векторной моделей будут подробно рассмотрены в дальнейших лекциях:

- WordNet в сочетании с векторной моделью информационного поиска в экспериментах H. Voorhees и P. Vossen,
- WordNet в булевской модели поиска вопросно-ответной системы Южного Методистского университета США,
- Традиционные информационно-поисковые тезаурусы в комбинации с разного рода статистическими моделями,

- Тезаурус для автоматического индексирования в булевских моделях поиска документов, в задаче автоматической рубрикации, автоматического аннотирования.

Интеграция разнородных баз данных

Под базой данных (БД) будем понимать коллекцию согласованных взаимосвязанных данных, которые имеют некоторое «скрытое (внутри) значение». БД похожи на базы знаний (БЗ), поскольку они также используются для описания некоторой предметной области, с целью хранения, обработки и доступа к необходимой информации о ней. Однако, есть и различия. Базы данных содержат (и способны обрабатывать) большие массивы относительно простой информации (при этом доступ возможен только к этим явно введенным данным). В базах знаний обычно хранится меньший объем информации, но она имеет более сложную структуру, что позволяет использовать возможности логического вывода и получать такие утверждения, которые не были в явном виде введены.

Подразумевается, что к любой БЗ могут быть применимы 3 операции: **определить (define)**, **сказать (tell)** (или сделать утверждение) и **спросить (ask).** Каждая из операций может использовать один или более собственных языков, например, язык описания схем и ограничений, язык обновления (для новых утверждений), язык запросов и язык ответов. Преимущества использования дескриптивной логики (DL) для улучшения каждого из языков широко изучены в литературе по базам знаний. Здесь будут рассмотрены 3 важных проблемы, возникающие при управлении данными: 1) выражение концептуальной модели предметной области (онтологии) для конкретного источника данных, 2) интеграция нескольких источников, 3) выражение и выполнение запросов. Для каждой из задач существует подход с использованием DL.

Сначала – несколько основных понятий из области проектирования БД и их использования.

В первую очередь, еще до создания БД, возникает необходимость описать предметную область таким языком, чтобы описание было понятно как обычным пользователям, так и разработчикам. Это описание выполняется на языке достаточно высокого уровня и имеет форму требований. В области БД таким языком является модель данных на основе ER-диаграмм (ER-модель). В рамках ER-модели окружающий мир представляется как набор сущностей, связанных n-арными (n-местными) отношениями и описанных атрибутами, имеющими атомарные значения. Полученная *семантическая модель* предметной области может храниться на компьютере, также как и сами данные, но, как правило, она содержит общую и постоянную

информацию (например, о том, что «отдел предприятия имеет ровно одного руководителя») в противоположность конкретным фактам («Петров является руководителем отдела поставок»). Эта семантическая модель вводит термины для описания предметной области и определяет их значения путем задания взаимосвязей и ограничений. Этот уровень представления данных наиболее близок онтологическому представлению.

Из семантической модели создается *логическая схема*, описывающая структуры данных в БД, типы данных, взаимосвязи и ограничения. Наиболее популярной и часто используемой для выражения логической схемы является реляционная модель данных. При спецификации логической схемы используются СУБД. В реляционной модели данные хранятся в таблицах (отношениях), содержащих строки (кортежи), которые в свою очередь состоят из ячеек со значениями простых типов данных (целые числа, строки, даты и т.п.). Поэтому для описания логической схемы требуется описать имена таблиц, имена их столбцов (атрибутов) и соответствующие типы данных. Реляционные СУБД требуют, чтобы в каждой таблице был определен набор столбцов (ключ) уникально идентифицирующий строку таблицы. Часто СУБД предлагают средства для поддержания целостности с помощью задания ограничений целостности – утверждений, которые способны отделить корректные состояния базы данных от некорректных.

БД используются для хранения информации о текущем состоянии «окружающего» мира. При этом делается, так называемое, «предположение о замкнутости». Это предположение означает, что факт является ложным, пока он явно не задан как истинный. Такое предположение работает хорошо, только если БД описывает очень ограниченную область. В частности в БД не разрешается описывать дизъюнктивную информацию и поддерживается ограниченная форма навешивания квантора существования: если нет информации о значении атрибута, то оно считается равным *null*.

Декларативный язык SQL для описания запросов к реляционным БД, определения содержимого таблиц и их структуры, является универсальным и мощным практическим средством. Однако, с точки зрения теории, язык логики предикатов первого порядка имеет более «элегантный» вид, если представить, что таблицы являются предикатами.

Например, формула

$$\exists m, d1, d2.\text{снабжает}(\text{'intel'}, \text{r}, m, d1) \wedge \text{снабжает}(\text{'intel'}, \text{r}, m, d2) \wedge (d1 \neq d2)$$

могла бы определять запрос относительно тех получателей (значения свободной переменной r), которые снабжались поставщиком ‘intel’ одним и тем же материалом m в различные моменты времени (d1 и d2).

СУБД скрывает от пользователя детали реализации тех или иных функций, делая прозрачным физический уровень: способ организации хранения данных на носителях информации, вспомогательные структуры данных для ускорения доступа и даже тот факт, что БД распределена по нескольким узлам сети. Но у пользователя может возникнуть желание получить информацию сразу из нескольких независимых (и, скорее всего, разнородных) источников (БД, файлы и пр.). В такой ситуации возникает проблема связывания различных логических схем в одну (предоставляемую пользователю).

Интеграция разнородных источников данных – фундаментальная проблема, возникшая в последние десятилетия перед сообществом разработчиков БД. Цель интеграции данных состоит в том, чтобы предоставить единый интерфейс к различным источникам и позволить пользователям сосредоточиться на определении того, что они хотят узнать. В результате интеграция должна освободить пользователя от поиска релевантных источников данных, взаимодействия с ними по отдельности, отбора и комбинирования данных из различных источников. Проектирование системы интеграции данных – очень сложная задача.

Рассмотрим «классические» подходы к ее решению. Первый из них состоит в использовании *федеративных* БД, которые независимо хранят одну и ту же информацию, периодически синхронизируя свои состояния. Для синхронизации n федеративных БД требуется определить $O(n^2)$ связей. Другой подход состоит в создании единого *централизованного хранилища данных.* Данные из разнородных источников периодически копируются в хранилище (требуется O(n) связей для n БД). Третий подход (наиболее эффективный, но и трудоемкий) использует технологию создания программных оболочек или *медиаторов* (mediators, wrappers), обеспечивающих единый интерфейс доступа к различным БД.

Задача, проектирование системы интеграции данных, состоит из нескольких подзадач. Онтологический подход может успешно применяться для решения двух подзадач:

1. Для спецификации содержимого разнородных источников данных в виде онтологии.
2. Для выполнения процесса ответов на запросы адресованных интегрирующей системе и основанных на спецификации источников.

Спецификация содержимого разнородных источников данных

Обычно архитектура системы интеграции данных позволяет явно моделировать данные и информационные потребности (т.е. определять те

данные, которые система предоставляет пользователю) на различных уровнях:

- *Концептуальный уровень* содержит концептуальное представление источников и согласованных интегрируемых данных вместе с явным декларативным описанием отношений между их компонентами.
- *Логический уровень* содержит представление источников в терминах логической модели данных.

Концептуальный уровень

Данный уровень содержит формальные описания понятий, отношений между понятиями и дополнительные информационный требования. Эти описания являются независимыми от системы интеграции и ориентированы на описание семантики приложения. На концептуальном уровне выделяют 3 элемента:

- Концептуальная схема уровня предприятия (понимаемого в широком смысле). Здесь представляются наиболее общие понятия связанные с приложением
- Концептуальная схема информационного источника является концептуальным представлением данных
- Концептуальная схема предметной области используется для описания объединения концептуальной схемы уровня предприятия и различных Концептуальных схем информационных источников. Кроме того сюда же входят дополнительные межсхемные отношения.

Элементарные понятия реляционной модели данных, такие как сущности, отношения и атрибуты, могут быть выражены на языке дескриптивной логики (DL) следующим образом. По ER-схеме S создается база знаний $\varphi(S)$, которая определяется так:

- множество атомарных концептов $\varphi(S)$ состоит из множества сущностей и доменных символов S;
- множество атомарных отношений $\varphi(S)$ получается из множества отношений и атрибутных символов S;
- множество аксиом включения $\varphi(S)$ состоит из утверждений формализующих иерархические зависимости между сущностями и отношениями, ограничения на сущности имеющие атрибуты и/или связанные отношениями и ограничения кардинальности для ролей каждого из отношений.

Важным свойством отображения $\varphi(S)$ является его взаимная однозначность, что позволяет строить логические выводы в рамках базы знаний средствами DL, а затем переносить результаты на ER-модель.

Таким образом, моделирование концептуальных схем БД при помощи онтологий не только возможно, но и дает дополнительные преимущества.

Наиболее интересным элементом на концептуальном уровне является схема предметной области, интегрирующая схему предприятия и схемы информационных источников, а также связывающая схемы информационных источников между собой. При этом используются *межсхемные отношения.* Они формально представляются как утверждения вида

$$L_i \subseteq_{ext} L_j ,$$
$$L_i \subseteq_{\text{int}} L_j ,$$

где L_i, L_j - являются некоторыми выражениями в различных схемах. Они могут быть либо отношениями с одинаковым числом аргументов, либо концептами. Первый индекс рядом со знаком включения означает, следующее: любой объект, удовлетворяющий выражению L_i в i-м источнике, удовлетворяет также выражению L_j в j-м источнике. Например, если известно, что множество сотрудников описанных в источнике 1 является подмножеством сотрудников описанных в источнике 2, то это может быть выражено так:

$$Сотрудник_1 \subseteq_{ext} Сотрудник_2$$

Второй тип межсхемных утверждений (соответствует индексу int) указывает, что концепт, описанный выражением L_i в i-м источнике, является подконцептом того, который описан выражением L_j в j-м источнике. Например, если известно, что концепт «сотрудник» описанный в источнике 1 является подконцептом концепта «личность», который описан в источнике 2, то это может быть выражено так:

$$Сотрудник_1 \subseteq_{\text{int}} Личность_2$$

Логический уровень

Данный уровень представляет описание логического содержимого для каждого источника, которое называется *схемой источника*. Обычно схема источника выражается в терминах набора отношений, используя логическую реляционную модель данных. Связывание логического представления источника и концептуальной схемы предметной области возможно двумя способами:

- подход на основе глобального представления. С каждым концептом концептуальной схемы предметной области ассоциируется запрос над отношениями источника. Таким образом, каждый концепт можно понимать как представление источника;
- подход на основе локального представления. Каждое отношение в источнике ассоциируется с запросом, описывающим его содержимое в терминах концептуальной схемы предметной области. Другими словами содержимое отношения внутри источника выражается концептами предметной области.

Для описания содержимого источника используется понятие *запрос*. Запрос определяется как дизъюнкция конъюнкций над множеством атомов. Каждый из атомов является концептом, отношением или атрибутом.

$$q(\vec{\mathbf{x}}) \leftarrow conj_1(\vec{\mathbf{x}}, \vec{\mathbf{y}}_1) \vee \cdots \vee conj_m(\vec{\mathbf{x}}, \vec{\mathbf{y}}_m).$$

Вопросы к лекции

1. Перечислите традиционные подходы к обработке запроса. В чем их недостатки?
2. Чем критерий полноты отличается от критерия точности?
3. Назовите способы улучшения поиска при помощи тезаурусов и онтологий.
4. Перечислите основные элементы ER-модели.
5. В чем на Ваш взгляд проявляется «интеллектуальность» агентов Semantic Web?

Литература

1. Jackson P., Mouliner I. Natural language processing for online applications: text retrieval, extraction, and categorization. John Benjamins Publishing Company. Amsterdam / Philadelphia. 2002.
2. Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. The Description Logic Handbook : Theory, Implementation and Applications

5. ОНТОЛОГИИ ПРЕДМЕТНЫХ ОБЛАСТЕЙ И ПРИКЛАДНЫЕ ОНТОЛОГИИ: НАЗНАЧЕНИЕ, ОТЛИЧИТЕЛЬНЫЕ ЧЕРТЫ, РЕШАЕМЫЕ ЗАДАЧИ (ПРИМЕРЫ ПРОЕКТОВ)

Онтология в области культуры CIDOC CRM

Краткое описание

CIDOC CRM (Conceptual Reference Model), представляющей собой формальную онтологию, предназначенную для улучшения интеграции и обмена гетерогенной информацией по культурному наследию. Более конкретно, CRM определяет семантику схем баз данных и структур документов, используемых в культурном наследии и музейной документации, в терминах формальной онтологии. Модель не определяет терминологию, появляющуюся в конкретных структурах данных, но имеет характерные отношения для ее использования.

Модель может служить, как руководством для разработчиков информационных систем, так и общим языком для экспертов предметной области и специалистов по информационным технологиям. Она предназначена для покрытия контекстной информации исторического, географического и теоретического характера об отдельных экспонатах и музейных коллекциях в целом.

Структурно CRM состоит из иерархии классов и широкого набора свойств (бинарных отношений), связывающих классы между собой. Все концепты (классы и свойства) модели можно разделить на три группы. Первая группа включает классы и отношения, охватывающие наиболее общие понятия окружающего мира: постоянные и временные сущности, отношения участия, зависимости, совпадения во времени. Вторая группа содержит понятия, частично поддерживающие функции управления: приобретение и учет единиц хранения, передача прав собственности на объекты культуры. К третьей группе относятся классы и свойства, используемые для внутренней организации самой онтологии: средства необходимые для подключения внешних источников терминов, например, тезаурусов по отраслям культуры.

Иерархия классов модели CRM делится на 2 ветви: Постоянные сущности и Временные сущности. Прочие классы являются вспомогательными.

На самых нижних уровнях иерархии классов появляются понятия характерные для сферы культуры: Хранение, Перемещение (ценностей), Проект или Процедура (в том числе техника производства), Период (в том числе художественный стиль). Иерархия классов может быть гибко

расширена, используя встроенный класс Тип. Наибольший интерес представляют свойства. Классы на нижних уровнях иерархии имеют около 10-15 свойств. Причем большая часть свойств наследуется от классов-предков. Названия свойств представляют собой глагольные фразы, выбранные так, что при последовательном связывании двух классов свойством получается осмысленная фраза с субъектом (первый, если считать слева направо, класс), предикатом (свойством) и объектом (второй класс).

Например, «(E29) Проект или Процедура (P68F)» обычно применяет «(E57) Материал» или «(E33) Лингвистический Объект (P72)» имеет язык «(E56) Язык»

На рис 13. – рис. 16 изображены различные части онтологии CRM (снимки экрана сделаны в редакторе Protege).

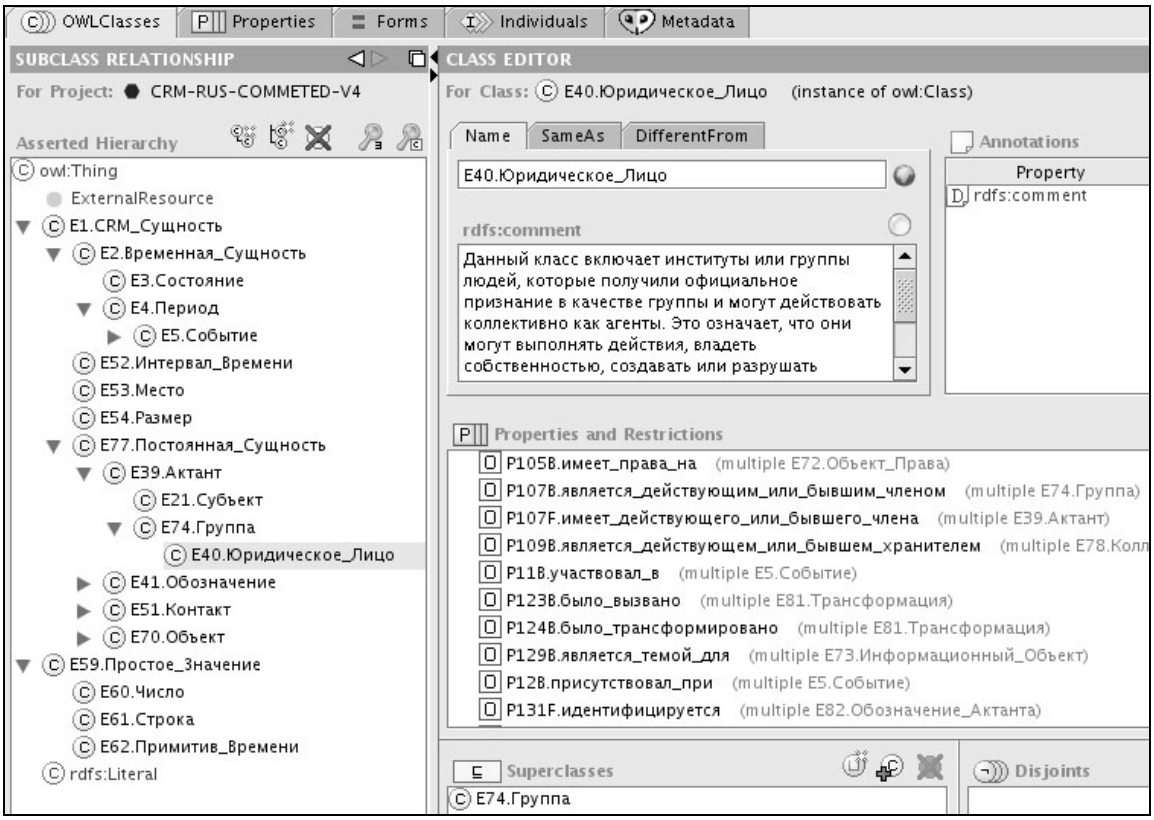

Рис. 13. Иерархия классов онтологии CRM (в левой части), текстовое описание и свойства класса «E40.Юридическое_Лицо» (в правой части окна). Здесь можно заметить, что большинство свойств класса “в глубине иерархии” являются унаследованными.

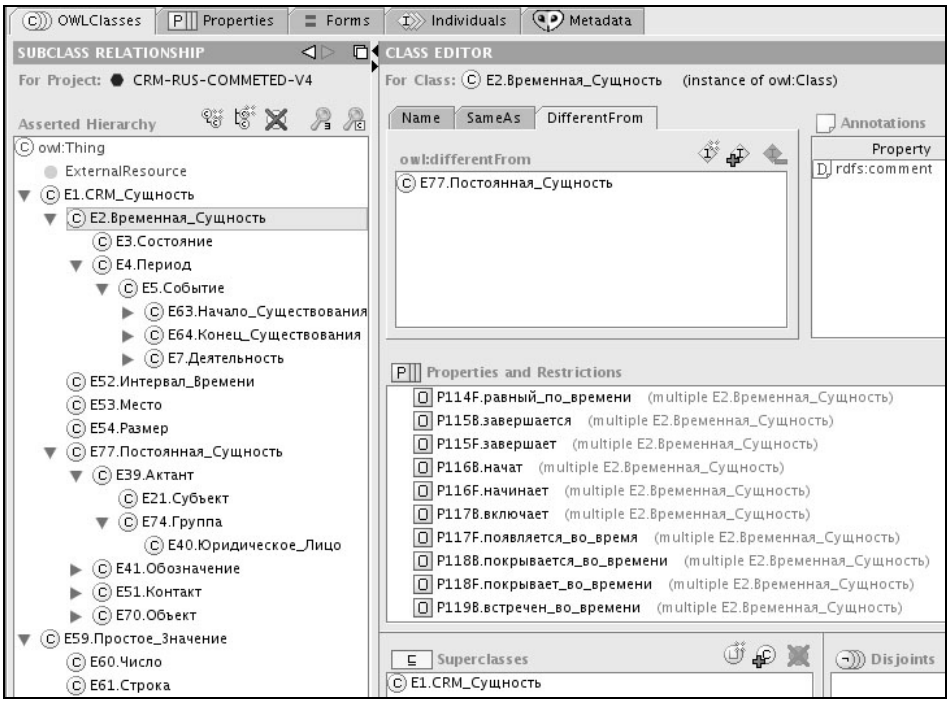

Рис. 14. Иерархия классов онтологии CRM. Свойства «на верхних уровнях иерархии» являются прямыми, а не унаследованными

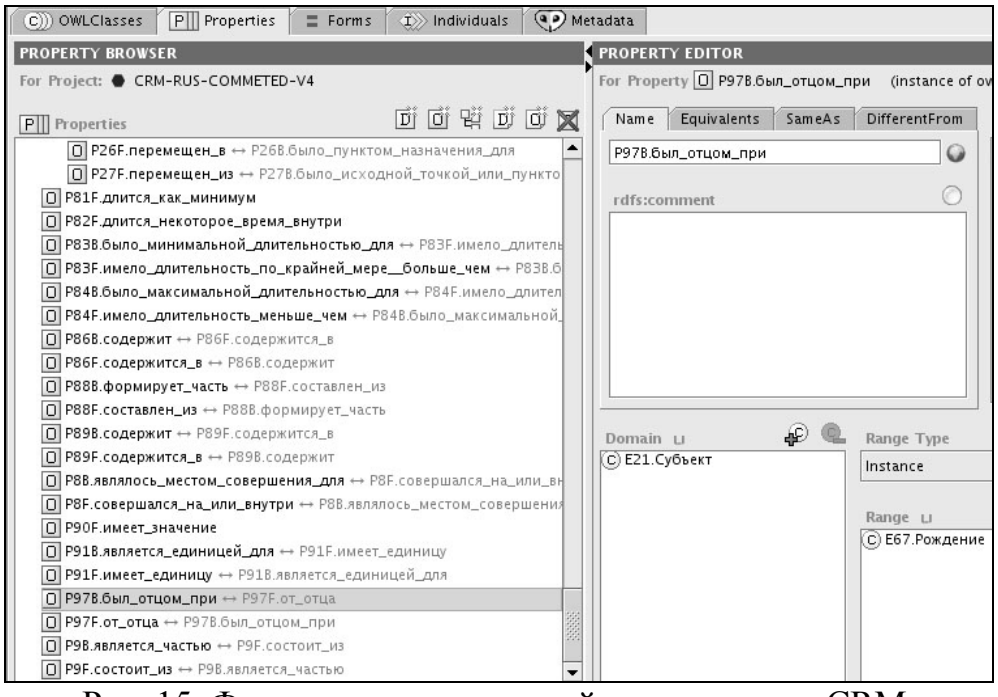

Рис. 15. Фрагмент иерархии свойств онтологии CRM. Свойство «P97B.был_отцом_при» связывает домен «E21.Субъект» и диапазон «E67.Рождение»

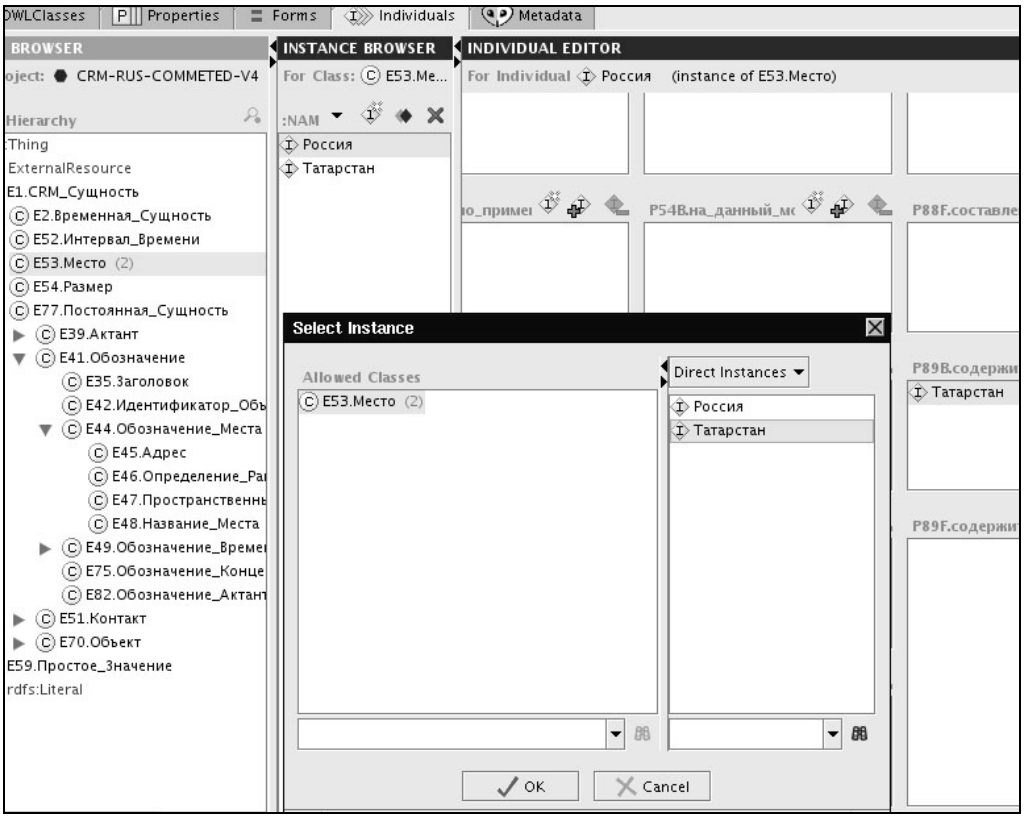

Рис. 16. Индивиды онтологии CRM.
В центре экрана изображено диалоговое окно для
выбора значения свойства «P89B.содержит»
индивида «Россия» класса «E53.Место».

Онтологии товаров и услуг

Одной из важных сфер применения онтологий является сфера предложения товаров и услуг.

К задачам, которые должны обеспечивать онтологии о товарах и услугах, относятся:

- Сбор информации о товарах
- Представление информации о товаре
- Классификация товаров – разделение по назначению
- Поиск по товарам
- Показ баннеров
- Показ текстов (обзоров, новостей, форумов) по товарам

К числу наиболее известных интернет-систем по товарам общего назначения являются такие системы как Froogle, Яндекс. Маркет, Тындекс.

Для классификации товаров традиционно используются классификации типа рубрикаторов, которые также рассматриваются как вид онтологической организации знаний.

Рубрикаторы как вид онтологий

Под **рубрикатором** понимается классификационная таблица иерархической классификации, содержащая полный перечень включенных в систему классов и предназначенная для систематизации информационных фондов, массивов и изданий, а также для поиска в них (ГОСТ 7.74-96).

Имеется главное теоретическое отличие терминов тезауруса от рубрик рубрикатора. Термины тезауруса являются фундаментально языковыми, в то время как рубрики соответствуют концептуальным категориям. Цель разработки информационно-поискового тезауруса – это найти хорошие, компактные слова и фразы для описания основных тем документов, сведя синонимы и квазисинонимы к дескрипторам тезауруса.

Цель создания рубрикаторов, которая не всегда достигается, но всегда ставится – это разработать совершенно отдельные концептуальные категории, которые взаимно не пересекаются. Идеально не должно быть пересечений между рубриками и не должно быть промежутков, то есть ни одна подобласть не должна остаться вне рубрик рубрикатора. Для достижения таких строгих целей рубрикатор структурируется, что может быть выполнено двумя основными способами – иерархической организацией рубрикатора и фасетной организацией рубрикатора.

Для того, чтобы определить рубрики достаточно строго и исключить пересечение значений, часто необходимо называть рубрики длинными и «неуклюжими» именами, например, «Тропические и субтропические фрукты и орехи; полевые культуры (Plantation crop). Такое словосочетание не встретить в тезаурусе, его назначение четко определить отдельную концептуальную категорию. Поскольку работать с такими сложно сформулированными сущностями достаточно тяжело, им обычно присваивается некоторая система классификационных кодов.

Еще одним следствием такого рода формулировок рубрик является то, что в таком виде их практически не встретить в реальных текстах, интернет-сайтах, поэтому если необходимо автоматизировать обработку перечней товаров, то необходимо каждой рубрике сопоставить наборы слов и словосочетаний, на основе которых нужно будет выводить данную рубрику.

OntoSeek - Content-Based Access to the Web (Guarino N., Проект – 1996г., Статья – 1999г.).

В качестве проекта, в рамках которой исследовался поиск по товарам на базе онтологий, рассмотрим систему OntoSeek.

OntoSeek – система, предназначенная для содержательного поиска в изданиях типа «желтые страницы» и каталогах товаров.

К числу особенностей системы относятся:

- использование произвольных естественно-языковых терминов для описания товаров и услуг;
- отсутствие ограничений на задание запросов на естественном в языке, базирующееся на семантической обработке запроса на основе онтологии,
- интерактивная помощь в формулировании запросов, в их обобщении и специализации.

В качестве представления информации о товарах были выбраны концептуальные графы. По сравнению с представлением вида атрибут-значение такие графы обеспечивают значительно более гибкий и более выразительный механизм представления запросов и описаний товаров. На базе концептуальных графов проблема сопоставления запроса и описания товара сводится к сопоставлению графов.

В качестве онтологии был взят WordNet, на основе описаний которого можно выявить синонимичность и родовидовые отношения слов.

Представление запросов основывается на графах, содержащих переменные. Так, если пользователь ищет автомобили, в внутри которых установлен радиоприемник, то запрос будет выглядеть следующим образом:

```
[<X> car]-> (part) -> [Radio].
```

Если пользователь ищет радиоприемник для автомобиля, то запрос представляется следующим выражением:

```
car]-> (part) -> [[<X> Radio].
```

Проблема использования такой онтологии как WordNet связана с тем, что в ней в явном виде не содержится информация о взаимной исключительности понятий.

Чтобы решить эту проблему, было предложено различать понятия-типы и понятия-роли и ввести следующие предположения:

- типы, которые не находятся в родовидовой иерархии, взаимно исключают друг друга;

- роли всегда подчиняются типам;
- роли, подчиняющиеся одному и тому же типу, рассматриваются как не взаимно исключительные, если это не указывается специально, например, отношением антонимии.

Примерами типов являются такие понятие как *человек* или *растение*, а примерами ролей такие понятия как *студент* или *ребенок.* Типы и роли различаются тем, что для типов принадлежность их примеров к своему типу является внутренне необходимым свойством, в то время как студент может перестать быть студентом, оставаясь все тем же человеком.

Вторым типом проблем является то, что верхние уровни WordNet являются слишком содержательно бедными для приложений, базирующихся на знаниях.

Отношения между понятиями, установленные на основе лингвистических критериев, не соответствуют отношениям между соответствующими классами объектов внешнего мира.

Вопросы к лекции

1. Что такое рубрикатор?
2. Использование рубрикаторов в интернет-системах по товарам и услугам
3. Система Ontoseek: какие проблемы пословного поиска и какими средствами предполагалось решать?

Литература

1. N. Guarino, C. Masolo, and G. Vetere: Ontoseek: Content-based Access to the Web, IEEE Intelligent Systems, Vol. 14, No. 3, pp. (www.loa-cnr.it/Papers/OntoSeek.pdf)
2. Цены и прайсы Рунета @ TYNDEX.RU (www.tyndex.ru)
3. Froogle (www.froogle.com)
4. Яндекс.Маркет (http://market.yandex.ru/)
5. The United Nations Standard Products and Services Code (www.unspsc.org)

6. ЯЗЫКИ ОПИСАНИЯ ОНТОЛОГИЙ. ОСНОВНЫЕ СИНТАКСИЧЕСКИЕ СТРУКТУРЫ: КЛАССЫ, ОТНОШЕНИЯ, АКСИОМЫ. ПРИМЕРЫ: RDF, OWL

6.1. Архитектура метаданных в World Wide Web

Документы, метаданные, связи

Когда вы переходите по ссылке URI, то получаете *нечто*. Мы будем называть это *нечто* ресурсом Сети. Часто под ресурсом понимается документ, поскольку в Сети много *читабельных* (удобных для чтения человеком документов – HTML страниц, PDF документов и т.п.). Иногда ресурс – это просто некий объект, когда полученный ресурс имеет машинопонятный вид или обладает скрытым внутренним состоянием.

В рамках этого раздела, термины ресурс, объект и документ являются синонимами.

Неотъемлемой характеристикой любого ресурса Сети является сопровождающая его информация. Эту «сверхинформацию» или информацию об информации (о ресурсе) принято называть метаданными.

Под метаданными будем понимать машинопонятную информацию о веб-ресурсах и других сущностях.

Термин «машинопонятная» является ключевым. Речь идет о понимании информации программными агентами. Причем «понимании» с одной целью – использовать информацию для решения задач возложенных на них (агентов) пользователем.

Метаданные должны иметь хорошо определенную ясную *структуру* и *семантику*.

Пример 1. Метаданные.

Объект, извлеченный из сети по протоколу http, может иметь дополнительную информацию (метаданные):

- *дата создания или дата прекращения действия*
- *владелец*
- *другая информация*

Таким образом, в Сети есть данные – ресурсы, есть метаданные – информация о ресурсах. Эта информация в свою очередь тоже может рассматриваться как данные (ресурс).

A1. Метаданные это – данные
(или информация об информации – это тоже информация)

Поскольку метаданные это данные, то они могут храниться в ресурсе (могут быть представлены как ресурс).

То есть любой ресурс Сети может хранить как данные, так и метаданные о себе или о других ресурсах.

На практике в Сети существует 3 способа передачи-получения метаданных.

1. метаданные хранятся и передаются внутри документа (тег HEAD в HTML, данные о документе MS Word.);
2. передача метаданных происходит во время HTTP-передачи при GET передаче от сервера к клиенту;
3. при POST или PUT передаче от клиента к серверу;
4. метаданные хранятся в каком-то другом документе.

Итак, метаданные могут храниться внутри документа, внутри другого документа, либо передаваться вместе с документом средствами HTTP.

Форма метаданных

Метаданные состоят из высказываний о данных и при представлении имеют форму имени (или типа высказывания) и набора параметров.

A2. Архитектура представляемая метаданными является набором независимых *высказываний (или утверждений)*.

Как следствие, при группировке 2-х и более высказываний об одном ресурсе они объединятся логическим «И». Альтернативные высказывания являются независимыми, а их наборы представляют собой неупорядоченные множества.

Конечно, высказывания можно комбинировать и другим способом, используя сложные синтаксические правила, но основной формой представления является неупорядоченный список, элементы которого связаны логическим «И».

Наиболее распространенной формой высказывания является следующая модель:

Ресурс – атрибут – значение

Здесь ресурс это – объект, о котором фиксируется высказывание, атрибут – некоторое свойство или параметр объекта, значение представляет некоторое значение из области значений атрибута (или *диапазона* значений атрибута данного объекта).

Пример 2. Использование модели «Ресурс – атрибут – значение».

E-mail – Date – 01.01.2006; E-mail – From – Vasya;

В общем виде высказывание может быть выражено так:

`(A u1 p q …),`

где A – имя (или идентификатор) для типа высказывания (Author, Date, …)

u1 – URI ресурса, о котором делается высказывание

p,q, … - другие параметры зависящие от типа высказывания, в том числе и представляющие значение атрибута.

Здесь можно провести аналогию с языками программирования.

В метаданных фиксирование высказывания можно сравнить с вызовом функции в процедурном языке.

В объектно-ориентированных языках программирования объект, для которого вызывается метод, имеет особое место среди других параметров (аргументов вызова). Для примера достаточно вспомнить ключевое слово “this” в C++. Также и в метаданных объект, о котором фиксируется высказывание, имеет особое место (u1).

В ООП набор методов (функций), которые можно вызвать для данного объекта, ограничен (интерфейсами или типом объекта). В метаданных набор типов высказываний, которые возможно сделать для данного объекта, потенциально не ограничен и определяется только выбором словаря.

Пространство имен атрибутов

Значения атрибутов и отношений могут сильно варьироваться, они могут задаваться спецификацией архитектуры или протокола. Но значения атрибутов могут быть определены для нужд одного конкретного приложения. Поэтому набор отношений и имен атрибутов должен быть легко расширяемым, а, следовательно, он должен быть расширяемым *децентрализовано*. Пространство URL подходит для определения имен атрибутов.

Пример 3. Словари с именами атрибутов.

1. *HTML элементы внутри элемента HEAD*
2. *Заголовки HTTP запроса уточняющие атрибуты объекта*
3. *(оба словаря: 1) и 2) определены внутри конкретных спецификаций)*

Связи

Отношение между двумя ресурсами будем называть связью.

Связь представляется тройкой

(A u1 u2)

A – тип отношения
U1 – URI первого ресурса
U2 – URI второго ресурса

Связи являются основой навигации в Сети. Они могут использоваться для построения структур внутри www, а также и для создания семантической Сети, в которой могут быть представлены знания об окружающем мире.

Иными словами, связи могут использоваться для определения структуры данных (в этом случае они являются метаданными), но они могут быть использованы и как форма представления данных.

Связи, как и прочие метаданные, могут быть переданы тремя (указанными выше) способами.

Одна из основных задач решаемых при проектировании архитектуры метаданных Сети состоит в том, чтобы сделать информацию самоописывающейся (self-describing).

Однако узким местом системы всегда является способ определения семантики метаданных и данных используемых внутри системы. Например, семантика метаданных заголовков e-mail и http сообщений определяется вручную на английском языке в виде спецификаций соответствующих протоколов. Эта семантика понятна людям (конечно, тем, кто знает английский). Чтобы теперь перейти к семантике понятной машине, нужно использовать подходящий логический язык или язык представления знаний. Тогда семантика (точное значение) некоторого высказывания может быть выражена в терминах других отношений (более абстрактных концептов логического языка).

Преимущество самоописывающейся информации состоит в том, что нет необходимости согласовывать значение каждого термина централизовано, стандартизировать семантику высказываний. Язык RDF позволяет описывать

метаданные о любых ресурсах Сети (и даже о сущностях находящихся за ее пределами).

RDF

RDF – язык представления информации о *ресурсах* WWW. В частности, RDF служит для представления *метаданных* связанных с ресурсами Сети, таких как заголовок, автор, дата последнего изменения страницы. Но RDF используется и для представления информации о ресурсах «второго типа», на которые можно только ссылаться (или идентифицировать в Сети при помощи URI), но к ним невозможно непосредственно получить доступ через Сеть.

Может оказаться что, в некоторых случаях для управления метаданными достаточно использовать XML и XML Schema (либо вообще ограничиться подэлементом HEAD элемента HTML). Но этот подход слабо масштабируется: при увеличении объема метаданных, усложнении их структуры управление метаданными построенными на основе XML Schema становиться трудоемкой задачей, для решения которой и предназначен RDF.

Модель данных RDF. RDF-граф

Базовой структурной единицей RDF является коллекция троек (или триплетов), каждый из которых состоит из субъекта, предиката и объекта (S,P,O). Набор триплетов называется RDF-графом. В качестве вершин графа выступают субъекты и объекты, в качестве дуг – предикаты (или свойства). Направление дуги, соответствующей предикату в данной тройке (S,P,O), всегда выбирается так, чтобы дуга вела от субъекта к объекту.

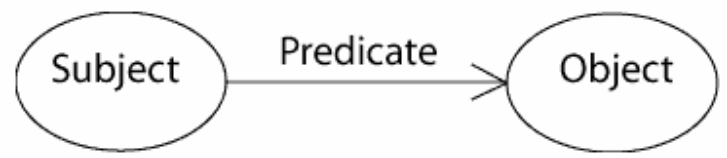

Рис. 17. RDF-тройка.

Каждая тройка представляет некоторое высказывание, увязывающее S, P и O.

Первые два элемента RDF-тройки (Subject, Predicate) идентифицируются при помощи URI.

Объектом может быть как ресурс, имеющий URI, так и RDF-литерал (значение).

RDF-литералы (или символьные константы)

RDF-литералы бывают 2-х видов: типизированные и не типизированные.

Каждый литерал в RDF-графе содержит 1 или 2 именованные компоненты:

- Все литералы имеют лексическую (словарную) форму в виде строки символов Unicode.
- Простые литералы имеют лексическую форму и необязательную ссылку на язык (ru, en…).

Типизированные литералы имеют лексическую форму и URI типа данных в форме RDF URI.

Замечание. Язык литерала не нужно путать с идентификатором локали. Языка относится только к текстам, написанным на естественном языке. Все трудности, возникающие при представлении данных на конкретном компьютере (при определении локали), должны решаться конечным пользователем.

Сравнение литералов

Два литерала равны тогда и только тогда, когда выполняются все перечисленные ниже условия:

1. Строки обеих лексических форм совпадают посимвольно;
2. Либо оба литерала имеют теги языка, либо оба не имеют;
3. Теги языка, если они имеются, совпадают;
4. Либо оба литерала имеют URI типа данных, либо оба не имеют;
5. При наличии URI типа данных, эти URI совпадают посимвольно.

Определение значения типизированного литерала

Приведем пример:

Пусть множество {T, F} - множество значений истинности в математической логике. В различных приложениях элементы этого множества могут представляться по-разному. В языках программирования {1, 0} (1 соответствует T, 0 соответствует F), либо {true, false}, либо {истина, ложь}.

Фактически задается некоторое отображение множества значений истинности на множество чисел или строк символов. Теперь значениями логического типа (bool или boolean) в становятся строковые значения или спецсимволы. Чтобы получить значения истинности необходимо воспользоваться обратным отображением.

Таким же образом происходит получение значения типизированного RDF литерала. За лексической формой стоит некоторое значение, которое определяется применением отображения. Это отображение определяется по URI типа данных и зависит от самого типа.

Вопросы к лекции

1. Почему AND-список высказываний о любом ресурсе может быть представлен неупорядоченным множеством?
2. Чем отличаются понятия ресурс, объект и документ в контексте Web?
3. Что такое RDF? Что представляет собой модель данных RDF и на чем она основана?

Литература

1. Uniform Resource Identifier (URI): Generic Syntax (http://tools.ietf.org/html/3986)
2. Аксиомы Web архитектуры: метаданные (http://www.w3.org/DesignIssues/Metadata.html)

Спецификации W3C

1. Понятия и абстрактный синтаксис RDF (http://www.w3.org/TR/rdf-concepts/)
2. Семантика RDF (http://www.w3.org/TR/rdf-mt/)
3. RDF Schema язык описания словарей RDF (http://www.w3.org/TR/rdf-schema/)
4. Руководство по OWL (http://www.w3.org/TR/owl-guide/)

6.2. Языки представления онтологий: RDFS, OWL. Язык запросов SPARQL

Языки, о которых пойдет речь в данном разделе, являются основными языками так называемой Семантической Сети (Semantic Web). О Semantic Web упоминалось ранее. Там же было отмечено, что на сегодняшний день наблюдается разрыв между способами представления метаданных (языками их определения) и теми интеллектуальными агентами, которые должны ими пользоваться. Языки описания метаданных и онтологий в Web, развиты очень хорошо, языки запросов и языки описания правил доведены до стадии технологических стандартов в данной области. Однако узким местом всё еще являются механизмы взаимодействия агентов на основе онтологий.

Многие популярные редакторы онтологий, которые будут описаны ниже, используют в качестве основного формализма дескриптивную логику (DL) и предоставляют средства для создания OWL-онтологий.

RDFS

Каждый из элементов триплета определяется независимо ссылкой на тип элемента и URI.

Предикат (в контексте RDF его обычно называют свойством) может пониматься либо как атрибут, либо как бинарное отношение между двумя ресурсами. Но RDF сам по себе не предоставляет никаких механизмов ни для описания атрибутов ресурсов, ни для определения отношений между ними. Для этого предназначен язык RDFS – (язык описания словарей для RDF). RDF Schema определяет классы, свойства и другие ресурсы.

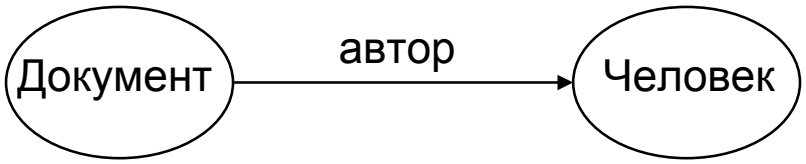

Рис.18. RDF-тройка субъект-предикат-объект

RDFS является семантическим расширением RDF. Он предоставляет механизмы для описания групп связанных ресурсов и отношений между этими ресурсами. Все определения RDFS выражены на RDF (поэтому RDF и называется «самоописывающимся» языком). Новые термины, вводимые RDFS, такие как «домен», «диапазон» свойства, являются ресурсами RDF.

Система классов и свойств языка описания RDF словарей похожа на систему типов объектно-ориентированных языков программирования, таких как Java. Но RDF отличается от большинства таких систем в том, что центральным аспектом является определение свойства, а не класса. Свойства в RDF определяются как пары (домен, диапазон). При этом домен представляет некоторое множество классов RDF, к которым данное свойство применимо, диапазон определяет допустимое множество ресурсов – значений свойства. Для сравнения в Java определение класса имеет законченную форму (свойства класса выражаются в полях и методах класса). В RDF, напротив, описание класса всегда остается открытым (набор свойств класса определяется вне самого класса).

Пример 1. Определим свойство «автор» с доменом «Документ» и диапазоном «Человек». В случае появления дополнительной информации о

свойствах «Документа», нет необходимости изменять описание класса «Документ». Достаточно добавить новое свойство с соответствующим доменом.

Пример «a-la RDF»:

```
Класс («Документ»);
Класс («Человек»);
Свойство («автор», «Документ», «Человек»).
```

Пример «a-la Java»:

```
Класс «Документ»
{
 «Человек» «автор»
}
```

Можно заметить, что при изменении смысла свойств изменять придется именно их. При этом все классы, зависящие от изменяемых свойств, косвенно изменят свою семантику.

Основное преимущество такого подхода в легкой расширяемости: добавление/удаление свойств интуитивно проще, чем управление множеством классов, обладающих каждый своим индивидуальным набором свойств (как в ООП). Фактически любой может расширять описание существующих ресурсов (лозунг Web: «Кто угодно может сказать что угодно о чем угодно!»).

Классы

Ресурсы могут объединяться в группы называемые *классами*. Члены класса (здесь наиболее близкий термин – «экземпляры» или «объекты» ООП) называются *экземплярами* класса. Сами классы также являются ресурсами, и идентифицируются ссылками RDF-URI. Для того чтобы указать, что ресурс является экземпляром класса, используется свойство rdf:type (“rdf” здесь используется как префикс пространства имен).

В RDF определения классов, свойств (*интенсионал*) отделены от множества экземпляров классов и значений свойств (т.н. *экстенсионала*). Так, два класса с одинаковыми экстенсионалами считаются различными, если они имеют разные наборы свойств (интенсионалы).

Экстенсионал и интенсионал

Рассмотрим множества

```
A = {0, 2, 4, 6, 8},
B = {x, | x = 2k, k = 0..4, k - целое},
C – множество неотрицательных четных чисел меньших 10.
```

В этом примере множество A полностью описывается своим экстенсионалом, множества B и C описываются интенсионалами, т.е. используя характеристические свойства данного множества. Множества, имеющие бесконечное число элементов могут быть описаны только своим интенсионалом. Однако при использовании интенсионала могут возникнуть парадоксы (например, парадокс Рассела: пусть множество M – множество всех множеств не содержащих себя. Содержит ли M само себя? Если содержит, то оно не удовлетворяет своему определению – интенсионалу, если M не содержит себя, то оно удовлетворяет определению и, следовательно, должно себя содержать). Для избежания подобных парадоксов в теории множеств вводятся специальные аксиомы. Примечательно, что RDF нарушает эти аксиомы. Классу RDF не запрещено быть экземпляром самого себя.

Группа ресурсов, являющихся классами, в RDFS описывается термином rdfs:Class

Над множеством классов определено отношение «подкласс-надкласс», описываемое RDFS свойством rdfs:subClassOf. Семантика данного отношения состоит в том, что экстенсионал любого подкласса данного класса (C), целиком включается (как множество) в экстенсионал данного класса (C). Другими словами, если (i) является экземпляром класса C*, а класс C* является подклассом класса C, то i является экземпляром класса C.

Любой класс RDFS по определению является подклассом самого себя.

В спецификации по RDFS определены также списки, коллекции и контейнеры ресурсов, текстовые пометки и комментарии для создания удобных для чтения примечаний к ресурсам.

РЕИФИКАЦИЯ (материализация, овеществление утверждений)

В случае, когда необходимо сделать утверждение об утверждении RDF, прибегают к так называемой реификации или материализации утверждений. Утверждение (или высказывание) выступает в роли объекта.

Для этого используется специальный класс rdf:Statement и его свойства rdf:subject, rdf:predicate и rdf:object. Каждое RDF утверждение является экземпляром класса rdf:Statement. По свойствам (и их значениям) можно однозначно идентифицировать само утверждение. Обладая этой информацией, возможно фиксировать утверждения об утверждениях.

Пример 2.

В базе данных электронного магазина хранится информация о том, что некий товар (Т) имеет цену х. Данное утверждение (1) (товар Т имеет цену х) может быть выражено Ивановым Иваном Ивановичем на языке RDF. Если далее потребуется высказать утверждение (2) о том, кто именно сделал утверждение (1), можно использовать механизм реификации (Рис.19).

```
Товар            Т                          #
rdf:Property     имеет цену                 #
Цена             х                          #
-------------------------------------------
rdf:Statement    Утверждение 1              *
rdf:subject      Т                          *
rdf:predicate    имеет_цену                 *
rdf:object       х                          *

rdf:Statement    Утверждение 1              +
rdf:Property     сделано_автором            +
Человек          Иванов Иван Иванович       +
```

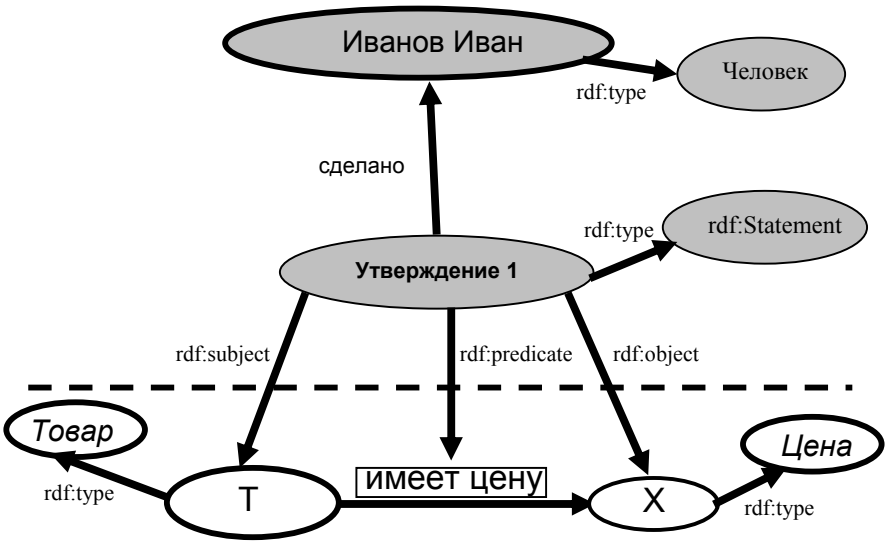

Рис. 19. Использование механизма реификации

Важный момент:

Фиксация только тех утверждений, которые помечены «*» не означает, что товар Т действительно имеет цену х. Даже вместе с утверждениями помеченными «+» вся информация, которую мы узнаём: «некто Иванов Иван

Иванович сделал утверждение о товаре Т, что он имеет цену х». Но не более того! Значение х цены товара Т фиксируется тройкой строк, помеченных «#».

Понятно, что новое утверждение (высказывание об Утверждении 1) также может быть подвергнуто реификации, поскольку синтаксически не отличается от Утверждения 1 (оно также является экземпляром класса rdf:Statement).

Далее приводится полный список классов и свойств RDF/RDFS.

Таблица 1. Классы RDFS

Имя класса	**Пояснение**
Rdfs:Resource	Класс ресурс, включает «всё».
Rdfs:Literal	Класс литеральных значений, текстовых строк или чисел.
rdf:XMLLiteral	Класс XML литералов
Rdfs:Class	Класс классов.
rdf:Property	Класс RDF свойств.
Rdfs:Datatype	Класс типов данных RDF.
rdf:Statement	Класс утверждений.
rdf:Bag	Класс неупорядоченных контейнеров.
rdf:Seq	Класс упорядоченных контейнеров.
rdf:Alt	Класс контейнеров-альтернатив.
Rdfs:Container	Класс RDF контейнеров.
rdfs:ContainerMembership Property	Класс свойств «членства» в контейнерах, rdf:_1, rdf:_2, ..., все они являются подсвойствами свойства member(член).
rdf:List	Класс RDF списков.

Таблица 2. Свойства RDFS

Имя свойства	**Пояснение**	**Домен**	**Диапазон**
rdf:type	Субъект является экземпляром класса.	rdfs:Resource	rdfs:Class
Rdfs:subClassOf	Субъект является подклассом класса.	rdfs:Class	rdfs:Class
rdfs:subProperty Of	Субъект является подсвойством свойства.	rdf:Property	rdf:Property
Rdfs:domain	Домен свойства сеюъекта.	rdf:Property	rdfs:Class

Имя свойства	Пояснение	Домен	Диапазон
Rdfs:range	Диапазон свойства субъекта.	rdf:Property	rdfs:Class
Rdfs:label	Человекочитаемое название субъекта.	rdfs:Resource	rdfs:Literal
Rdfs:comment	Текстовое описание ресурса	rdfs:Resource	rdfs:Literal
Rdfs:member	Член ресурса субъекта.	rdfs:Resource	rdfs:Resource
rdf:first	Первый элемент списка.	rdf:List	rdfs:Resource
rdf:rest	Оставшийся за первым элементом «хвост» списка.	rdf:List	rdf:List
Rdfs:seeAlso	Дополнительная информация о субъекте.	rdfs:Resource	rdfs:Resource
Rdfs:isDefinedBy	Определение ресурса субъекта.	rdfs:Resource	rdfs:Resource
rdf:value	Свойство, используемое для структурированных значений	rdfs:Resource	rdfs:Resource
rdf:subject	Субъект RDF утверждения (см. реификация).	rdf:Statement	rdfs:Resource
rdf:predicate	Предикат утверждения (см. реификация).	rdf:Statement	rdfs:Resource
rdf:object	Объект RDF утверждения (см. реификация).	rdf:Statement	rdfs:Resource

Возможности и ограничения языка RDF (RDF Schema)

Сам по себе RDF не является стандартом метаданных как, например, DublinCore, FOAF, vCard.

Все что он «умеет» это – фиксировать утверждения о ресурсах, их свойствах и значениях этих свойств.

Важные свойства языка:

- обобщенный способ работы с метаданными;
- ориентация на программное обеспечение в качестве конечного потребителя информации;
- возможность осуществлять автоматическую обработку Web-ресурсов:
    - поиск;
    - каталогизацию;
    - генерацию иерархических карт сайтов

Недостатки RDF

Открытость и расширяемость RDF ведет к тому, что «кто угодно (т.е. любой пользователь RDF) может сказать что угодно (т.е. фиксировать произвольное утверждение) о чем угодно (т.е. о любом ресурсе)», используя RDF. RDF не запрещает делать бессмысленных утверждений или утверждений, не согласующихся с другими. Следовательно, нет никакой гарантии целостности и непротиворечивости RDF-описаний. Вся ответственность за проверку ложится на получателей (конечных пользователей) метаданных, т.е. на разработчиков приложений обрабатывающих RDF данные.

Способы представления RDF-описаний

Ниже приводится пример двух способов представления RDF графов: в форме XML документа (часто более удобной для автоматической обработки) и в форме последовательностей троек – так называемый N Triple или N3 синтаксис (удобный для восприятия человеком).

XML синтаксис

```
    <?xml version="1.0"?>
    <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:exterms="http://www.example.org/terms/">

    <rdf:Description
rdf:about="http://www.example.org/index.html">
    <exterms:creation-date>August 16, 1999</exterms:creation-
date>
    </rdf:Description>

    <rdf:Description
rdf:about="http://www.example.org/index.html">
    <dc:language>en</dc:language>
    </rdf:Description>

    </rdf:RDF>
```

N3 синтаксис (удобный для чтения человеком, и расширяющий исходную модель данных RDF)

```
    <ex:index.html> <dc:creator>              exstaff:85740 .
    <ex:index.html> <exterms:creation-date>   "August 16, 1999" .
```

```
<ex:index.html> <dc:language>              "en".
```

На этих примерах можно заметить «тяжеловесность» XML синтаксиса RDF, по сравнению с N3-синтаксисом. Но он более удобен для *сериализации* RDF-графов.

Из вышесказанного о RDF и метаданных можно сделать вывод, что RDF имеет довольно слабые (по объему выразительные средства), не основан на каком-либо логическом формализме. Это язык описания метаданных, причем метаданных в широком смысле слова: имеющих произвольную структуру и смысл. Пожалуй, единственный принцип, которому следует RDF это основной лозунг Web. RDF – универсальный инструмент, и поэтому требует настройки для решения конкретных специализированных задач. Способ такой «настройки» состоит в расширении RDF при помощи словарей. Перейдем к рассмотрению одного из расширений RDF для области проектирования и представления онтологий.

OWL

OWL (Web Ontology Language) – язык представления онтологий в Web. Фактически это словарь, расширяющий набор терминов определенных RDFS. OWL-онтологии могут содержать описания классов, свойств и их экземпляров. Создание OWL это – ответ на необходимость представления знаний в Сети в едином формате. Исторически предшественником OWL был язык DAML+OIL, объединивший 2 инициативы: проект DAML (DARPA Agent Markup Language) и проект OIL (Ontology Inference Layer). Наиболее ранним проектом представления онтологий в Web был SHOE (Simlpe HTML Ontology Extensions). Ветви развития языков описания онтологий для Web показаны на рис. 3. Верхний уровень: OIL, DAML+OIL и OWL продолжает развиваться, но наибольшей популярностью пользуется OWL.

OWL с 2004 года является рекомендацией W3C и объединяет лучшие черты свих предшественников.

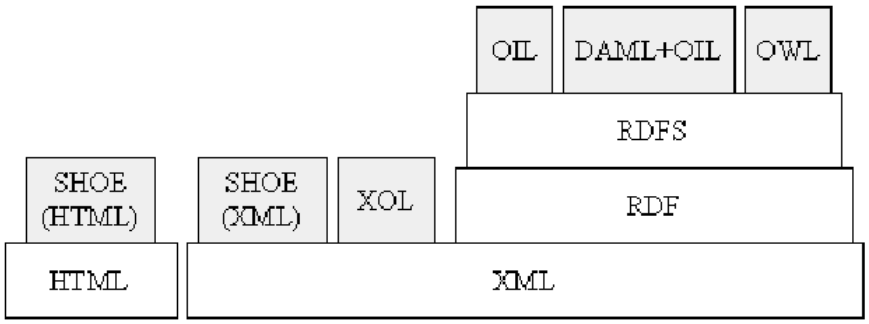

Рис.20. Основные ветви развития языков описания онтологий для Web

Язык OWL имеет 3 диалекта (подмножества терминов):

- OWL Lite – имеет наименьшую выразительную мощность из всех, но для решения простых задач его может быть достаточно.
- OWL DL – обладает выразительной мощью эквивалентной дескриптивной логике (разрешимой части логики предикатов первого порядка). Для большинства задач встречающихся при проектировании онтологий выразительности этого диалекта достаточно. Диалект имеет два важных свойства: полнота (все заключения являются вычислимыми) и разрешимость (все вычисления завершаются в конечное время). В частности в OWL DL классу запрещено быть экземпляром.
- OWL Full – наиболее выразительный диалект. Эквивалентен RDF. При использовании OWL Full нет никаких гарантий по вычислимости заключений.

Каждый из этих диалектов (кроме Lite) является расширением предыдущего и как следствие: любая OWL Lite онтология является OWL DL онтологией, а любая OWL DL онтология является OWL Full онтологией.

Структура OWL-онтологии

Любая онтология имеет заголовок и тело. В заголовке содержится информация о самой онтологии (версия, примечания), импортируемых онтологиях. За заголовком следует тело онтологии, содержащее описания классов, свойств и экземпляров.

Базовые элементы OWL

Классы

В OWL введен новый термин класс (owl:Class). Необходимость этого объясняется тем, что не все классы диалектов DL и Lite являются rdfs классами (в этом случае owl:Class является подклассом rdfs:Class). В диалекте Full, подобных ограничений нет, и owl:Class фактически является синонимом rdfs:Class.

Для организации классов в иерархию используется свойство rdfs:subClassOf.

Особое место занимают два взаимодополняющих класса Thing и Nothing. Первый их них является надклассом любого класса OWL, второй – подклассом любого класса OWL. Экземпляр любого класса OWL входит в

экстенсионал класса Thing. Экстенсионал класса Nothing является пустым множеством.

OWL класс может быть описан 6 способами:

1. идентификатором класса (URI)
2. перечислением всех экземпляров класса
3. ограничением на значение свойства
4. пересечением 2 и более определений классов
5. объединением 2 и более определений классов
6. дополнением (логическим отрицанием) определения класса

Только первый способ определяет именованный класс OWL. Все оставшиеся определяют анонимный класс через ограничение его экстенсионала. Способ 2 явно перечисляет экземпляры класса, способ 3 ограничивает экстенсионал только теми экземплярами, которые удовлетворяют данному свойству. Способы 4-6 используют логические операции (AND, OR и NOT) над экстенсионалами соответствующих классов, чтобы определить экстенсионал нового класса.

Описания класса формируют строительные блоки для определения классов через аксиомы.

Простейшая аксиома определяющая именованный класс:

```
<owl:Class rdf:ID="Human"/>
```

Все что постулирует эта аксиома – существование класса с именем “ Human”.

В OWL определены еще 3 конструкции, комбинируя которые можно определять более сложные аксиомы классов.

- rdfs:subClassOf говорит о том, что экстенсионал одного класса (подкласса) полностью входит в экстенсионал другого (надкласса).
- owl:equivalentClass говорит о том, что экстенсионалы двух классов совпадают.
- owl:disjointWith говорит о том, что экстенсионалы двух классов не пересекаются. Иногда говорят, что таким образом определяются дизъюнктивные классы.

Свойства

В OWL выделяют две категории свойств: свойства-объекты (или объектные свойства) и свойства-значения. Первые связывают между собой

индивиды (экземпляры классов). Вторые связывают индивиды со значениями данных. Оба класса свойств являются подклассами класса rdf:Property.

Для определения новых свойств как экземпляров owl:ObjectProperty или owl:DatatypeProperty используются аксиомы свойств.

Пример аксиомы:

```
<owl:ObjectProperty rdf:ID="hasParent"/>
```

Все что постулирует данная аксиома – существование некоторого свойства "hasParent" связывающего экземпляры класса owl:Thing друг с другом.

Кроме того, OWL поддерживает следующие конструкции для построения аксиом свойств:

- Конструкции RDF Schema: rdfs:subPropertyOf (определяет подсвойство данного свойства), rdfs:domain (определяет *домен* свойства) и rdfs:range (определяет *диапазон* свойства)
- Отношения между свойствами: owl:equivalentProperty (определяет *эквивалентное свойство*) и owl:inverseOf (определяет *обратное свойство*)
- Ограничения глобальной кардинальности: owl:FunctionalProperty (определяет *однозначное свойство* – однозначное отображение домена свойства на диапазон) и owl:InverseFunctionalProperty (*обратно функциональное свойство*, т.е. определяет, что свойство обратное данному свойству является однозначным)
- Логические характеристики свойства: owl:SymmetricProperty (определяет свойство как *симметричное*) и owl:TransitiveProperty (определяет *транзитивное свойство*).

Индивиды (экземпляры классов или свойств)

Индивиды определяются при помощи аксиом индивидов (т.н. фактов). Рассмотрим два вида фактов:

1. факты членства индивидов в классах и о значениях свойств индивидов;
2. факты идентичности/различности индивидов

Пример аксиом индивида первого вида:

```
<Балет rdf:ID="ЛебединоеОзеро">
  <имеетКомпозитора rdf:resource="#Чайковский"/>
</Балет>
```

Данная аксиома постулирует сразу 2 факта: (1) существует некоторый индивид класса «Балет» имеющий имя «ЛебединоеОзеро»; (2) этот индивид связан свойством «имеетКомпозитора» с индивидом: «Чайковский» (определенным где-то в другом месте).

Первый факт говорит о членстве в классе, второй – о значении свойства индивида.

Аксиомы второго вида необходимы для суждения об идентичности индивидов. Дело в том, что в OWL не делается никаких предположений ни о различии, ни о совпадении двух индивидов, имеющих различные идентификаторы URI.

Подобные утверждения выражаются аксиомами идентичности с помощью следующих конструкций:

- owl:sameAs постулирует, что две ссылки URI ссылаются на один и тот же индивид.
- owl:differentFrom постулирует, что две ссылки URI ссылаются на разные индивиды.
- owl:AllDifferent предоставляет средство для определения списка попарно различных индивидов.

На рис. 21 проиллюстрированы основные элементы OWL-онтологии.

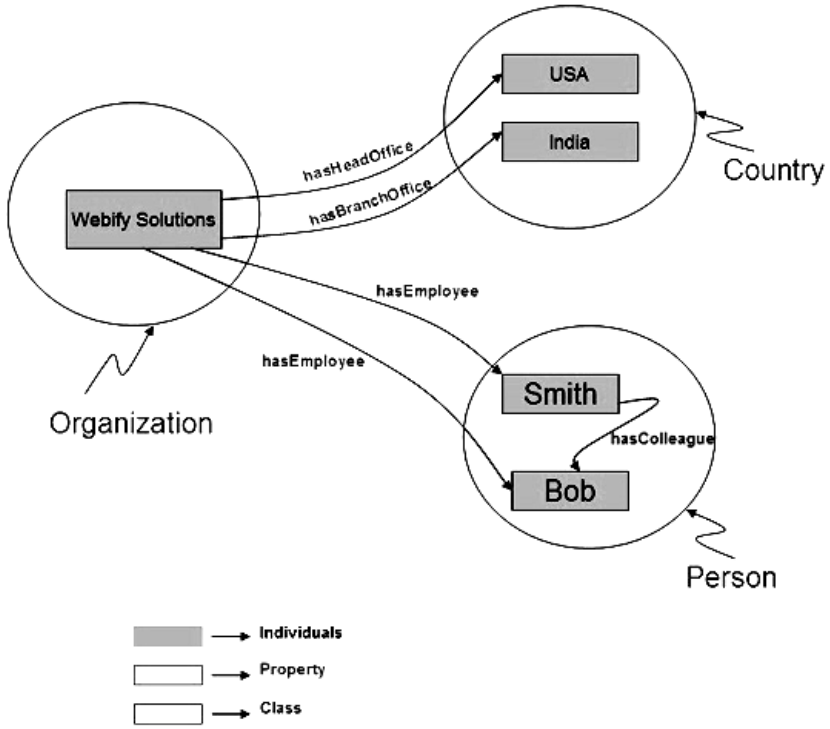

Рис. 21. Основные структурные единицы OWL-онтологии

SPARQL

Вероятно, сами по себе языки представления онтологий не были бы так сильно востребованы, если бы не возникало необходимости автоматически обрабатывать онтологии, наполнять их содержимым и *выполнять к ним запросы*.

Наиболее популярными среди языков запросов к RDF-хранилищам на сегодняшний день являются языки RDQL и SPARQL.

Рассмотрим несколько упрощенный синтаксис SPARQL-запроса :

```
SELECT     <v_list>
FROM  <onologyURI>
WHERE {    <template_list>.
           FILTER <filter_expr>
      }
```

- v_list – список имен переменных
- onologyURI – URI-ссылка на онтологию
- template_list – список шаблонов
- filter_expr – ограничения на значения переменных

Допустим, онтология содержит следующие RDF-триплеты:

```
(Foo1, category, "Total Members");
(Foo1, rdf:value, 199);
(Foo2, category, "Total Members");
(Foo2, rdf:value, 200);
(Foo2, category, "CATEGORY X");
(bar, category, "CATEGORY X");
(bar, rdf:value, 358).
```

Проследим за ходом выполнения запроса (имена переменных предваряются знаком «?»)

```
SELECT ?cat ?val WHERE {?x rdf:value ?val. ?x category ?cat.
FILTER (?val>=200).}
```

Семантика запроса:

выдайте все объекты cat предиката category, субъект которого (x) является также субъектом предиката rdf:value со значением val, не меньшим 200. Вместе со значениями cat выдать соответствующие значения val.

Ход выполнения запроса:

На место переменной x могут быть подставлены Foo1, Foo2 и bar (из исходной онтологии), причем Foo2 может быть подставлен дважды, поскольку имеет два свойства category).

При подстановке Foo1 значение переменной val не удовлетворяет ограничению в предложении FILTER. Во всех остальных случаях все условия запроса выполнены. Ниже представлен результат выполнения запроса.

Результат выполнения запроса: 3 пары значений (cat, val)

```
[
["Total Members", 200],
["CATEGORY X", 200],
["CATEGORY X", 358]
]
```

Вопросы к лекции

1. Для чего нужен RDFS?
2. Что такое реификация?
3. Чем отличается класс RDFS от класса OWL?

Основная литература

1. Uniform Resource Identifier (URI): Generic Syntax (http://tools.ietf.org/html/3986, 2005)
2. Аксиомы архитектуры Web: метаданные (http://www.w3.org/DesignIssues/Metadata.html, 1997)

Рекомендации W3C:

1. Понятия и абстрактный синтаксис RDF (http://www.w3.org/TR/rdf-concepts/, 2004)
2. Семантика RDF (http://www.w3.org/TR/rdf-mt/)
3. RDF Schema язык описания словарей RDF (http://www.w3.org/TR/rdf-schema/, 2004)
4. Руководство по OWL (http://www.w3.org/TR/owl-guide/, 2004)

Дополнительная литература

1. Декер С., Мельник С., Хермелен Ф., Фенсел Д., Клейн М., Брукстра Д., Эрдманн М., Хоррокс Я. Semantic Web: роли XML и RDF // Открытые система #9/2001.
2. Balani N. The future of the Web is Semantic // URL: http://www-128.ibm.com/developerworks/library/wa-semweb/index.html, 2005
3. Михаленко П. Язык онтологий в Web // Открытые системы #02/2004.
4. OWL, язык веб-онтологий. Руководство // URL: http://sherdim.rsu.ru/pts/ semantic_web/REC-owl-guide-20040210_ru.html, 2004

5. Ландэ Д. Семантический веб: от идеи – к технологии. URL: http://poiskbook.kiev.ua/sw.html, 2005.
6. Shadbolt N., Berners-Lee T., Hall W. The Semantic Web Revisited // IEEE Intelligent Systems 21(3) pp. 96-101, May/June 2006.

7. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРОЕКТИРОВАНИЯ ОНТОЛОГИЙ

7.1. Редакторы онтологий

Введение

При создании онтологий (как и при проектировании программного обеспечения или написании электронного документа) целесообразно пользоваться подходящими инструментами. Будем называть инструментальные программные средства, созданные специально для проектирования, редактирования и анализа онтологий, редакторами онтологий.

Основная функция любого редактора онтологий состоит в поддержке процесс формализации знаний и представлении онтологии как *спецификации* (точного и полного описания).

В большинстве, современные редакторы онтологий предоставляют средства «*кодирования*» (в смысле описания) формальной модели в том или ином виде. Некоторые дают дополнительные возможности по анализу онтологии, используют механизм логического вывода.

В этой части пособия будут описаны наиболее общие характеристики редакторов и проведен их сравнительный анализ. Подробно рассматривается редактор Protégé.

Общие характеристики редакторов

1. Поддерживаемые редактором формализмы и форматы представления.

Под *формализмом* понимается теоретический базис, лежащий в основе способа представления онтологических знаний. Примерами формализмов могут служить логика предикатов (First order logics - FOL), дескриптивная логика, фреймовые модели (Frames), концептуальные графы и т.п. Формализм, используемый редактором, может существенно влиять не только на внутренние структуры данных, но и определять формат представления или даже пользовательский интерфейс.

Формат представления онтологии задает вид хранения и способ передачи онтологических описаний. Под форматами подразумеваются языки представления онтологий: RDF, OWL, KIF, SCL.

Таким образом, некоторая формальная модель представляется в формализме FOL и может быть выражена средствами языка KIF.

Редакторы онтологий обычно поддерживают работу с несколькими формализмами и форматами представления, но часто только один формализм является «родным» (native) для данного редактора.

Функциональность

Важной характеристикой является функциональность редактора, т.е. множество сценариев его использования.

Базовый набор функций обеспечивает

- Работу с одним или более проектами:
    - Сохранение проекта в нужном формализме и формате (экспорт);
    - открытие проекта;
    - импорт из внешнего формата;
    - редактирование метаданных проекта (в широком смысле: от настройки форм редактирования и представления данных, до поддержки версий проекта)
- Редактирование онтологии. Набор возможных действий обычно включает создание, редактирование, удаление понятий, отношений, аксиом и прочих структурных элементов онтологии, редактирование таксономии.

К дополнительным возможностям редакторов относят поддержку языка запросов (для поиска нетривиальных утверждений), анализ целостности, использование механизма логического вывода, поддержка пользовательского режима.

Редактор Protégé.

С момента его создания Protégé многие годы использовался экспертами в основном для концептуального моделирования в области медицины.

В последнее время его стали использовать в других предметных областях. В частности при создании онтологий для Semantic web.

Изначально единственной моделью знаний поддерживаемой Protégé была фреймовая модель. Этот формализм сейчас является «родным» для редактора, но не единственным.

Protégé имеет открытую, легко расширяемую архитектуру и помимо фреймов поддерживает все наиболее распространенные языки представления знаний (SHOE, XOL, DAML+OIL, RDF/RDFS, OWL). Protégé поддерживает модули расширения функциональности (plug-in). Расширять Protégé для

использования нового языка проще, чем создавать редактор этого языка «с нуля».

Модель знаний Protégé

Protégé основан на модели представления знаний OKBC (Open Knowledge Base Connectivity). Основными элементами являются классы, экземпляры, слоты (представляющие свойства классов и экземпляров) и фасеты, задающие дополнительную информацию о слотах.

Пользовательский интерфейс

На рис.22 приведен снимок экрана показывающий общий вид редактора.

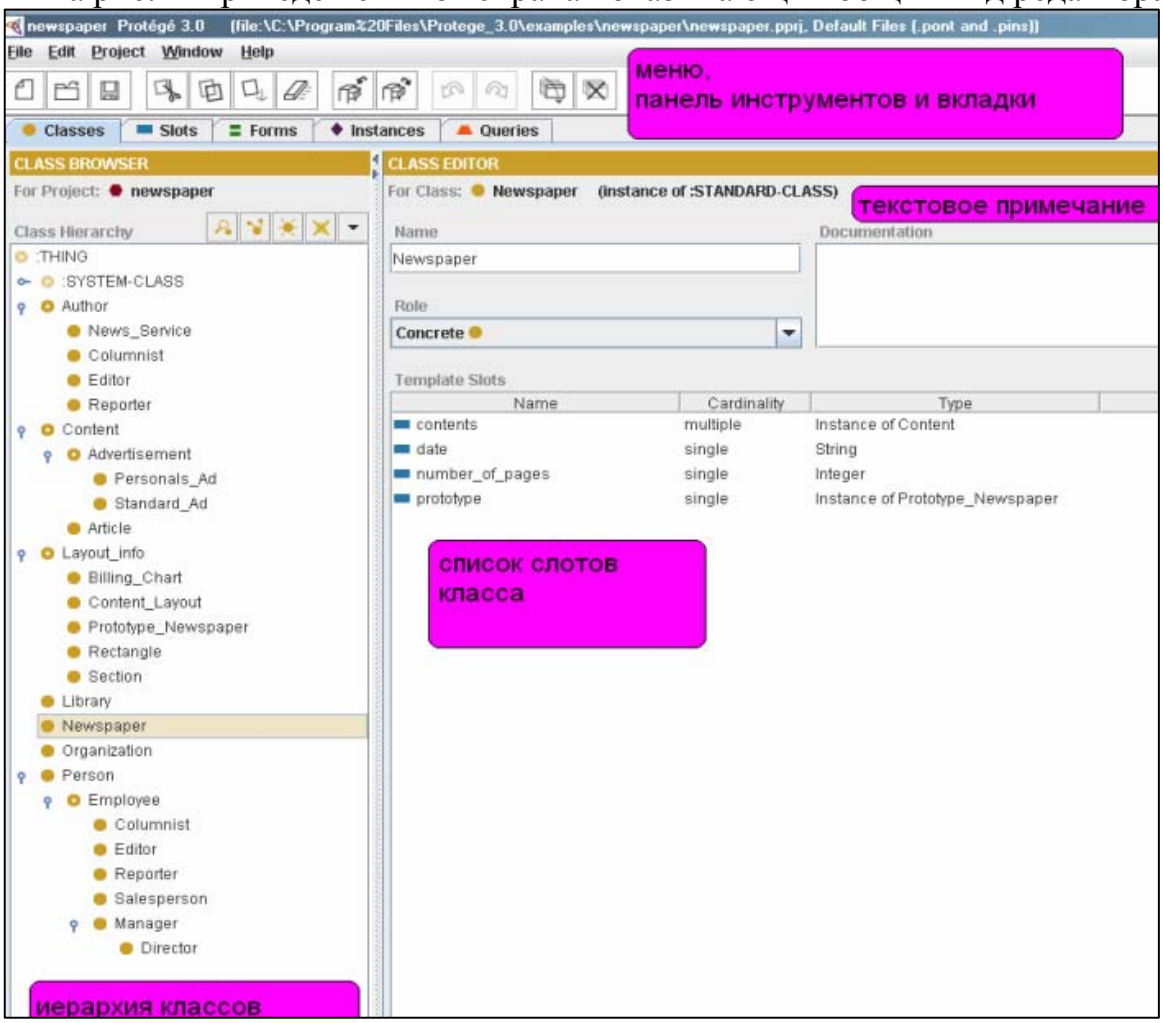

Рис.22 Общий вид редактора Protégé. Вкладка «Классы»

Пользовательский интерфейс состоит из главного меню и нескольких вкладок для редактирования различных частей базы знаний и ее структуры. Набор и названия вкладок зависит от типа проекта (языка представления) и может быть настроен вручную. Обычно имеются следующие основные вкладки: Классы, Слоты (или Свойства для OWL), Экземпляры, Метаданные.

Назначение основных вкладок – предоставить набор форм для заполнения базы знаний.

Классы

Функции: создание классов, слотов для данного класса, отображение иерархии классов, добавление текстовых примечаний к классам, поиск класса по шаблону.

Слоты

Функции: создание слотов, назначение домена и диапазона для данного слота, отображение иерархии и свойств слотов, добавление текстовых описаний слотов, поиск слота по шаблону, задание ограничений на значения слота.

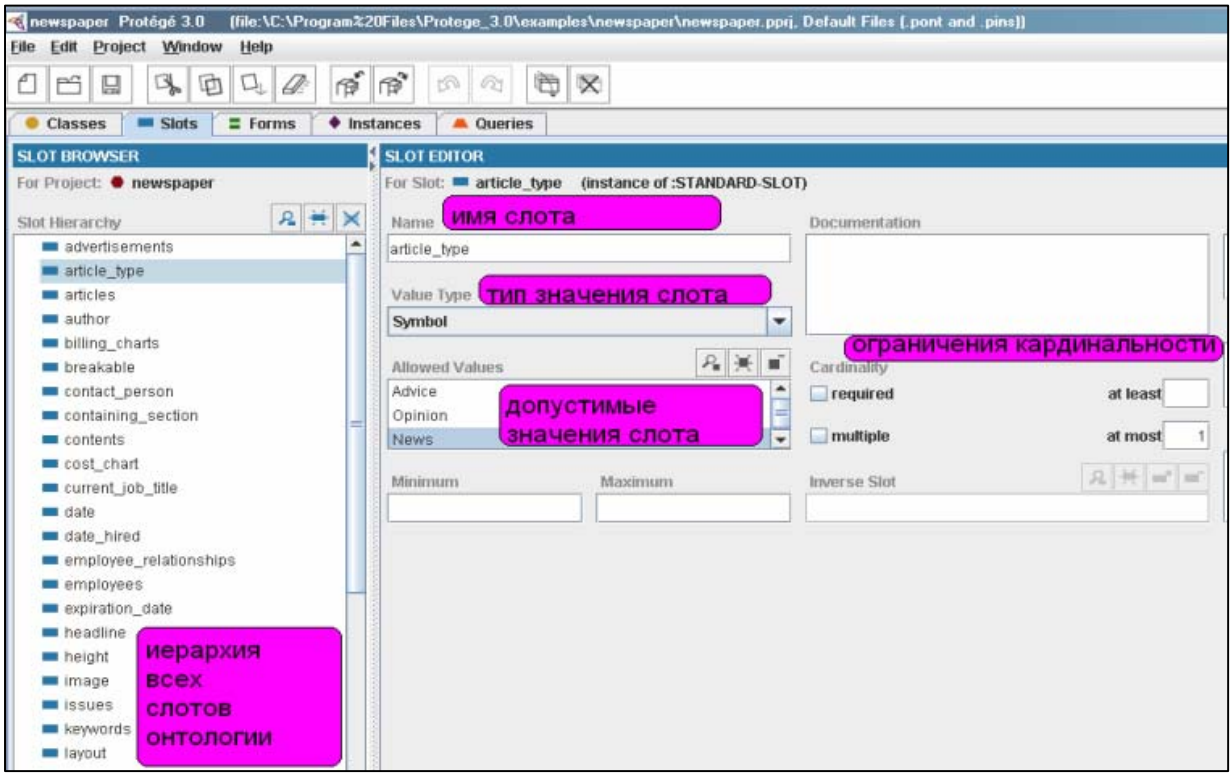

Рис.23. Вкладка «Слоты»

Экземпляры

Функции: создание экземпляров данного класса, отображение и редактирование свойств экземпляра, отображение иерархии классов, связывание экземпляров слотами, добавление текстовых описаний слотов, поиск слота по шаблону, задание ограничений на значения слота.

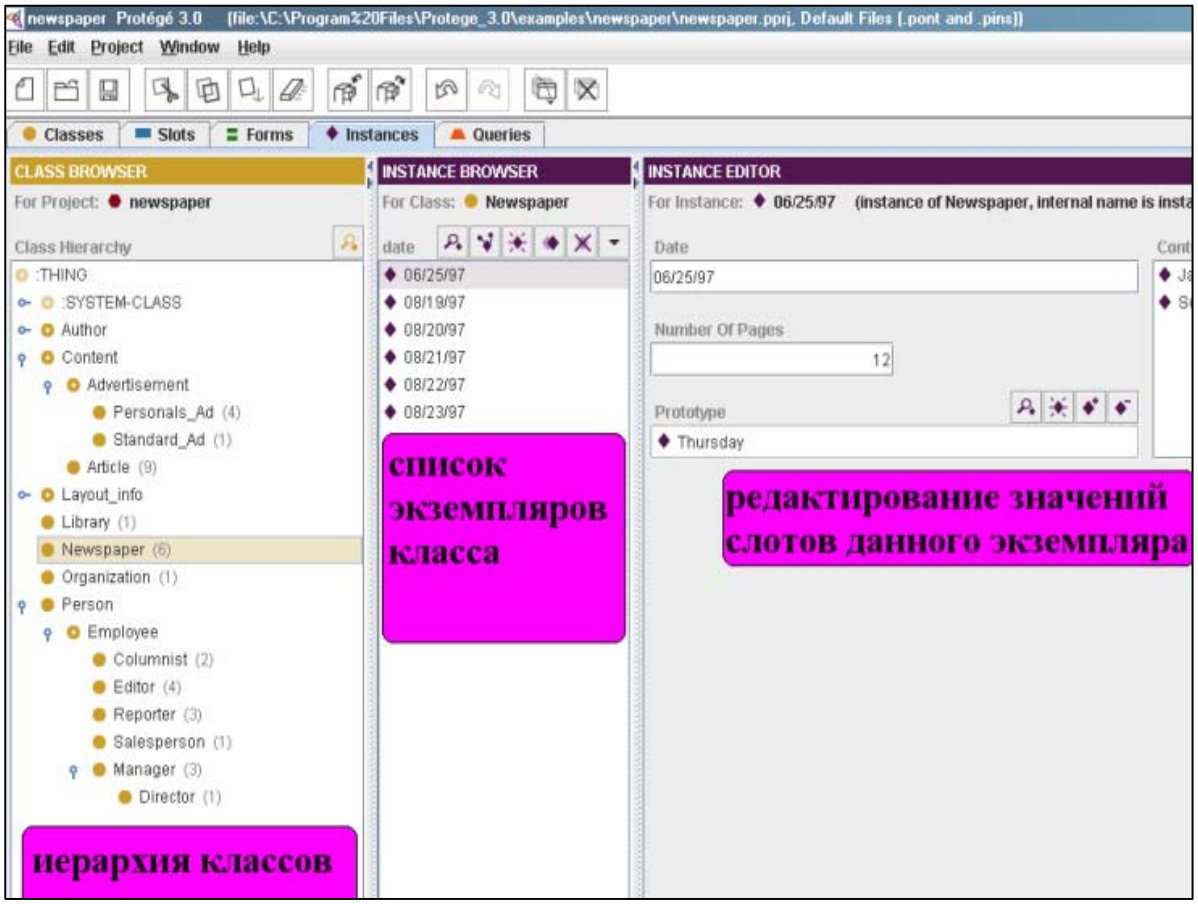

Рис.24. Вкладка «Экземпляры»

Сравнение редакторов

В последнее время количество общедоступных редакторов онтологий «перевалило» за 100. Но редко можно встретить универсальное и в то же время полезное средство. Ниже приводится таблица, описывающая основные характеристики наиболее популярных редакторов онтологий.

Таблица. Сравнение редакторов онтологий.

Имя	Описание	Формализмы, языки, форматы	URL
Ontolingua	совместная разработка	OKBC, KIF	www.ksl.stanford.edu/software/ontolingua/
WebOnto	совместный просмотр	OCML	kmi.open.ac.uk/projects/webonto/
Protégé	Создание, просмотр онтологии	база данных JDBC, UML, XML, XOL, SHOE, RDF и RDFS, DAML+OIL, OWL.	protege.stanford.edu
OntoSaurus	Web-браузер баз знаний на языке LOOM	LOOM	www.isi.edu/isd/ontosaurus.html
WebODE	Создание, методология Methontology	FLogic	webode.dia.fi.upm.es/WebODEWeb/index.html
OntoEdit	Разработка и поддержка онтологий	F-Logic, RDF-Schema и OIL	www.ontoknowledge.org/tools/ontoedit.shtml
OilEd	Поддержка логического вывода	DAML+OIL	oiled.man.ac.uk

Вопросы к лекции

1. Перечислите известные Вам редакторы онтологий.
2. Какой формализм является основным для редактора Protege?

Литература

1. Обзор инструментов инженерии онтологий -- О.М. Овдей, Г.Ю. Проскудина // Электронные библиотеки 2004.
2. Creating Semantic Web Contents with Protégé-2000 -- Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W. Fergerson, and Mark A. Musen, Stanford University // IEEE Intellegent Systems.

8. ЛИНГВИСТИЧЕСКАЯ ОНТОЛОГИЯ WORDNET

8.1. WordNet: Описание ресурса

(http://www.cogsci.princeton.edu/~wn/,
http://www.cogsci.princeton.edu/cgi-bin/webwn)

Лингвистический ресурс WordNet разработан в Принстонском университете США. WordNet относится к классу лексических онтологий. WordNet свободно доступен в Интернет и на его основе были выполнены тысячи экспериментов в области информационного поиска.

WordNet версии 2.1 охватывает приблизительно 155 тысяч различных лексем и словосочетаний, организованных в 117 тысяч понятий, или совокупностей синонимов (synset), общее число пар лексема – значение образует 200 тысяч. В состав словаря входят лексемы, относящиеся к четырем частям речи: прилагательное, существительное, глагол и наречие. Лексемы различных частей речи хранятся отдельно и описания, соответствующие каждой части речи, имеют различную структуру. Между существительными в словаре установлены следующие семантические отношения:

- синонимия;
- антонимия;
- гипонимия / гиперонимия - отношение, которое иначе может быть названо ВЫШЕ-НИЖЕ, ISA - отношение. Отношение транзитивно и несимметрично. Гипоним наследует все свойства гиперонима. Это отношение является центральным отношением для описания существительных.
- меронимия (отношение ЧАСТЬ-ЦЕЛОЕ). Внутри этого отношения выделяются отношения *быть_элементом* и *быть_сделанным_из*.

WordNet: гипонимы

X рассматривается как гипоним У-а если из того, что из любого утверждения «А – это Х», следует, утверждение «А – это У», а из утверждения «А – это У» не следует утверждение «А – это Х»

Например,

«Это – собака», значит «Это-животное». - «Это – животное» не следует «это собака»

«Это – жеребец», значит, «Это – лошадь». «Это – лошадь» не следует «это - жеребец».

«Это – человек, который идет», значит «Это- человек, который двигается».

«Это – человек, который двигается», не следует «Это- человек, который идет».

Отношение Часть-Целое

Меронимия представляет собой скорее совокупность несколько отличающихся отношений, чем четкое отделяемое отношение.

В качестве первого определения меронимии, которое однако исключает некоторые очевидные случаи отношения часть-целое, может служить следующее положение:

X является меронимом Y тогда и только тогда, если предложения вида *Y имеет X (или Xы) и X – это часть Y* являются нормальными для X и Y, интерпретируемых как родовые понятия.

Отношение ЧАСТЬ-ЦЕЛОЕ представляет собой семейство близких отношений. Наиболее центральным типом этого отношения представляют физические объекты.

Сущности такие как группы, классы и коллекции состоят в отношении меронимии со своими элементами.

Примеры групп: *племя, команда, комитет, семья, оркестр, суд, отряд и др*.

Примеры классов: *пролетариат, аристократия, буржуазия*.

Примеры коллекций: *куча, лес, библиотека (как коллекция книг)*

Если и ЦЕЛОЕ и ЧАСТЬ являются неисчислимыми, то говорят об отношении ингредиентов, например, *заварной крем и молоко.*

Если ЦЕЛОЕ – исчислимое, а часть неисчислимое, то это так называемое отношение объект-материал: *бокал - стекло*.

Если ЦЕЛОЕ неисчислимое, а ЧАСТЬ – исчислимое, то говорят об отношении вещество-частица: *песок – песчинка, снег – снежинка, дождь – капля*.

Примеры описания частей (включая подвиды) в WordNet:

собственно часть

```
flower, bloom, blossom -- (reproductive organ of
angiosperm plants esp. one having showy or colorful
parts)
     PART OF: angiosperm, flowering plant -- (plants
     having seeds in a closed ovary)
```

элемент

```
homo, man, human being, human -- (any living or
extinct member of the family Hominidae)
     MEMBER OF: genus Homo -- (type  genus of the
     family Hominidae)
```

вещество

```
glass -- (a brittle transparent solid with irregular
atomic structure)
     SUBSTANCE OF: glassware, glasswork -- (articles
     made of glass)
     SUBSTANCE OF: plate glass, sheet of glass --
     (glass formed into a thin sheet)
```

Описание прилагательных

Прилагательные делятся на качественные прилагательные и относительные. Семантическое описание качественных прилагательных значительно отличается от описания других основных категорий слов и базируется не на отношении гипонимии, а на отношении антонимии. Важность этого отношения для качественных прилагательных проявляется в психолингвистических тестах: когда человека просят назвать ассоциацию на качественное прилагательное, он чаще всего называет его антоним. Например, самая частая ассоциация на слово good (хороший) – это слово bad (плохой) и наоборот.

Важность антонимии для организации качественных прилагательных становится понятной, если учесть, что функцией этих прилагательных является выражение величин атрибутов и эти атрибуты обычно являются биполярными. Антонимичные прилагательные обычно выражают противоположные полюса атрибута.

Описание глаголов

Для описания глаголы были разделены на семантические поля. На первом этапе были отделены глаголы, обозначающие действия и события, от глаголов, обозначающих состояния. Первая группа глаголов была разделена на 14 семантических полей: глаголы движения, восприятия, контакта, коммуникации, соревнования, изменения, познания, потребления, создания, эмоций, обладания, ухода за телом, и глаголы, относящиеся к социальному поведению.

Границы между группами являются достаточно расплывчатыми. Например, многие глаголы не могут однозначно расклассифицированы как

глаголы познания или коммуникации (*wonder, speculate, confirm, judge* и др.). Также, например, глагол whistle в предложении «The bullet whistled past him" может классифицироваться и как глагол издания звука и как глагол движения. Если такие глаголы представлять как однозначные, они должны относиться к более чем одному семантическому полю. В WordNet глаголы чаще описывались как полисемичные, если обнаруживалось, что они могут быть отнесены одновременно к разным семантическим полям.

Основу описания иерархии глаголов составляют отношения тропонимии. Кроме того, описываются отношения причина-следствие (каузальные отношения) и другие отношения лексического следствия.

При разработке таких больших структурных ресурсов как WordNet обычно возникает ряд проблем, таких как

- проблема параллельных мест, когда одно и то же явлению описывается несколькими независимыми друг от друга понятиями (синсетов),
- проблема качественных путей, когда каждое отношение похоже на правду, а более длинный путь устанавливает «странные» отношения.

Одной из серьезных проблем, приводящих к неправильным путям иерархии является проблема установления таких отношений, когда вышестоящее понятие частично характеризует нижестоящее. Часто это связано с проблемой смешения понятий-типов и понятий-ролей.

Смешение типов и ролей

Одной из серьезных проблем, приводящих к неправильным путям иерархии является проблема установления таких отношений, когда вышестоящее понятие частично характеризует нижестоящее. Часто это связано с проблемой смешения понятий-типов и понятий-ролей.

Так, например, N.Guarino критикует отношения в WordNet: Человек всегда живое существо, но он (она) начинает играть роль каузального агента только в некоторых ситуациях. Та же проблема возникает для яблока, которое всегда плод растения, и в некоторых ситуациях может быть пищей. Проблема в том, что человек и яблоко – это типы сущностей, в то время как каузальный агент и пища – это роли.

Один из аргументов в пользу различения типов и ролей в лингвистических онтологиях – это то, что они различаются в способах наследования свойств. WordNet не различает эти два типа понятий и помещает их в одни и те же иерархии.

В соответствии с онтологическими подходами понятия-типы не должны находиться в иерархиях ниже понятий-ролей. Более радикальный подход заключается в том, чтобы разделить иерархии типов и ролей.

EuroWordNet

Ресурс WordNet, разработанный для английского языка, вызвал в мире огромный интерес к разработке такого рода ресурсов для десятков других языков.

Проекты разработки ворднетов для разных языков в рамках проекта включали два этапа. На первом этапе (1996-1999) ворднеты создавались для голландского, испанского и итальянского языков. На втором этапе – для французского, чешского, немецкого и эстонского языков.

В проекте стоял серьезный выбор, нужно ли стремиться к разработке языково-независимой структуры, с которой необходимо сопоставить единицы каждого языка, или может быть нужно иметь единую систему синсетов – новая единица в иерархической сети может быть включена, если хотя бы один язык из рассматриваемых имеет лексему или устойчивый оборот с таким значением.

По принятому в проекте решению каждый ворднет должен сохранять специфику своего языка. При этом каждый ворднет должен содержать отсылки на значения английского ворднета, что позволяет сравнивать ворднеты, обнаруживать непоследовательности в построении ворднетов и видеть различия в устройстве языковых систем.

Одновременно в рамках проекта была создана небольшая онтология верхнего уровня, к которой должен был приписан каждый создаваемые ворднет**.**

Вопросы к лекциям

1. Как называются элементарные структурные единицы WordNet?
2. Перечислите основные отношения в WordNet.
3. Какими средствами в WordNet представляются глаголы?

Литература

1. http://www.cogsci.princeton.edu/~wn/
2. http://www.illc.uva.nl/EuroWordNet/

3. Азарова И.В., Митрофанова О.А., Синопальникова А.А. Компьютерный тезаурус русского языка типа WordNet // Компьютерная лингвистика и интеллектуальные технологии. Труды Международной конференции Диалог'2003 (Протвино, 11-16 июня 2003 г .) М., 2003. С . 43-50.
4. *Азарова И.В., Синопальникова А.А., Яворская М.В.* Принципы построения wordnet-тезауруса RussNet // Компьютерная лингвистика и интеллектуальные технологии. Труды Международной конференции Диалог'2004 (“Верхневолжский”, 2-7 июня 2004 г .) М., 2004. С. 542-547.
5. Поляков В.Н. Проект WordNet и его влияние на технологии компьютерной и когнитивной лингвистики (Обзорная статья) // Труды Казанской школы по компьютерной и когнитивной лингвистике TEL-2002. – Казань, 2002. С.6-61
6. Fellbaum C. Parallel Hierarchies in the Verb Lexicon. In Proceedings of ‘The Ontologies and Lexical Knowledge bases’ workshop (OntoLex 2002).
7. Gangemi, Aldo, Nicola Guarino, and Alessandro Oltramari, 2001. Conceptual analysis of lexical taxonomies:the case of wordnet top-level. In *Proceedings of the international conference on Formal Ontology in Information Systems*. ACM Press.
8. Miller G., Beckwith R., Fellbaum C., Gross D. and Miller K., Five papers on WordNet. - CSL Report 43. Cognitive Science Laboratory, Princeton University, 1990.
9. Miller G. 1998 Nouns in WordNet. In: Fellbaum, C (ed) WordNet – An Electronic Lexical Database. – The MIT Press. pp.23-47.
10. Vossen P. Tutorial Wordnet*, EuroWordNet and Global WordNet*, International Conference RANLP – 2003 (Recent Advances in Natural Language Processing), Borovets, Bulgaria, 10-12 September 2003
11. Wilks, Yorick, 2002. Ontotherapy: or how to stop worrying about what there is. Invited presentation, Ontolex 2002, Workshop on Ontologies and Lexical Knowledge Bases,27th May. Held in conjunction with the Third International Conference on Language Resources and Evaluation - LREC02, 29-31 May, Las Palmas, Canary Islands.

8.2. WordNet: Применение в информационном поиске

Для того, чтобы попытаться реализовать схему автоматического концептуального индексирования и концептуального поиска необходимо иметь лингвистический ресурс, организованный на основе понятий или значений слов. Поэтому такие ресурсы как WordNet могут использоваться как база для организации приложений концептуального индексирования и поиска. В этой лекции рассматривается два таких эксперимента.

Using WordNet for Text Retrieval (Ellen M. Vorhees)

Целью экспериментов была попытка выполнить поиск документов на основе не отдельных слов, а значений WordNet. Для каждого документа сначала выполняется процедура разрешения многозначности существительных (см. п.), которая единственное значение и в результате которой каждому тексту ставится в соответствие вектор синсетов WordNet. После того, как вектор создан, с ним могут выполняться такие же операции, как и с пословными векторами.

Эффективность использования векторов синсетов сравнивалась с эффективностью информационного поиска на основе стандартной вектора слов. В стандартном прогоне и документы, и запросы представляются как вектора лемм всех значимых слов. В концептуальных прогонах, документы и запросы представляются как вектора, состоящие из трех подвекторов: вектор лемм слов, не найденных в WordNet, или тех, многозначность которых не удалось разрешить – например, относящихся к другим частям речи; вектор синсетов для слов с разрешенной многозначностью, и леммы для слов с разрешенной многозначностью. Второй и третий подвектора представляют собой альтернативные представления документа, поскольку одни и те же слова этого документа порождают отдельные элементы каждого вектора.

Для экспериментов было использовано 5 разных коллекций документов (компьютерная область, медицинская область, газетные статьи и др.), и для каждой коллекции было выполнено более 30 различных запросов.

Для каждого запроса стандартный прогон векторной модели сравнивался со следующими комбинациями подвекторов:

- 110 – данная комбинация дает одинаковые веса словам, отличным от существительных и синсетам существительных;
- 211 – данная комбинация учитывает как синсеты, так и леммы существительных, оставшиеся слова поэтому учитываются в двойном размере;
- 101 – в данной комбинации подвектор синсетов игнорируется, а существительные и другие леммы документа получают одинаковые

веса. Однако этот вектор отличается от стандартного прогона, поскольку результат сравнения для системы подвекторов высчитывается как сумма результатов сравнения каждого вектора.

Оценки эффективности информационного поиска на основе показателя средней точности показали серьезной ухудшение эффективности для векторов, включающих синсеты (от 6.2 до 42.3%).

Основная причина такого ухудшения эффективности заключается в том, что процедура разрешения многозначности для слова в запросе может выбрать одно значение, а для того же слова в документе другое значение. Например, при поиске по запросу ”separation anxiety in infants and preschool children” стандартный прогон выдает 7 релевантных документов в первых 15 документах, в то время как прогон 110 выдает только один релевантный документ в первых 15 документах. Проблема в выборе значения слова separation, для которого в WordNet описано 8 значений. Процедура разрешения многозначности выбирает такое значение этого слова в запросе, которое не было выбрано ни в одном из релевантных текстов.

Другая группа экспериментов по использованию WordNet в информационном поиске исследовала возможность расширения запроса синонимами или другими словами, связанными со словами запроса отношениями, описанными в WordNet. В таких экспериментах нет необходимости выбора единственного значения слова, что в случае ошибки приводит к серьезному ухудшению результатов поиска.

Для экспериментов были использованы следующие соображения.

Во-первых, расширяться должны только важные для запроса понятия. Важность аппроксимируется количеством документов, в которых встречается конкретное слово запроса - слова, частотность которых в документах коллекции больше некоторого числа N, не участвуют в расширении запроса.

Во-вторых, чтобы смоделировать разрешение многозначности запрос расширяется только теми словами, которые оказались в окрестностях расширения по крайней мере двух слов запроса.

Таким образом, сначала для каждого слова запроса, частотность которых меньше некоторого числа N, и каждого синсета для значений этого слова извлекается список близких по WordNety слов.

Те слова, которые встретились по крайней мере в двух таких списках добавляются к исходному запросу.

Исследовались различные величины N – 10% коллекции и 5% коллекции.

Для расширения запроса использовались синсеты, находящиеся на расстоянии 1 и 2 отношения от исходных синсетов - все виды связей трактовались одинаково.

Добавленные слова могли учитываться с разными величинами весов w=0.3, 0.5, 0.8.

Максимальное улучшение, которое удалось получить – 0.7% средней точности, что не является статистически значимой величиной (N=5%, расстояние – 2, w=0.3).

Авторы подчеркивают, что идея аппроксимации разрешения многозначности путем поиска повторов в списках расширения не является удачной, поскольку чаще всего это решение приводило к добавлению в запрос очень общих слов, таких как «система».

Для того, чтобы исключить из рассмотрения эффект лексической многозначности и исследовать возможности WordNet по расширению поискового запроса, были выполнены эксперименты с ручным выбором значения многозначных слов в запросе.

Для каждого синсета, соответствующего слову запроса, в запрос могут быть добавлены разные слова, на основе различных отношений данного синсета: например, синонимы, все слова из нижестоящих синсетов иерархии гипоним - гипероним, все слова, отстоящие на один шаг от текущего синсета.

Чтобы исследовать все такие возможности был образован вектор, состоящий из 11 подвекторов: 1 для слов исходного запроса, один для синонимов, 1 для каждого типа отношений существительных в WordNet. Сходство с документами вычислялась как взвешенная сумма результатов сравнений с каждым из подвекторов.

Исследовались четыре варианта векторов:

1) расширение только по синонимам,
2) расширение синонимы + полная иерархия вниз
3) расширение синонимы+ родители+ полная иерархия вниз
4) расширение синонимы+ слова из любых синсетов на один шаг по любому типу отношений.

Тестирование проходило на двух типах вопросов: более длинной и более короткой версии запросов. При поиске по полному запросу ни одной из комбинаций не удалось улучшить результаты поиска более чем на 2 процента. Короткие вопросы состояли из небольшого списка синсетов, например, {cancer}, {skin_cancer}, {phramaceutical}.

Для укороченного запроса, используя тип расширения 4), при котором все добавления учитывались с коэффициентом 0.5 было получено 35% улучшение: средняя точность для укороченного запроса без расширения была

– 0.1634, с расширением – 0.2205. Средняя точность поиска по полному запросу – 0.3586.

Выводы автора заключаются в том, что для успешного применения WordNet в информационном поиске необходимо значительно улучшить эффективность автоматического расширения лексической многозначности, между тем парадигматических отношений в WordNet недостаточно для решения этой задачи.

Проект Meaning

Проект Meaning является продолжением проекта EuroWordNet. Авторы проекта мотивируют необходимость продолжения работ тем, что десятки человек-лет были затрачены для создания ворднетов для разных языков, но этих усилий недостаточно, чтобы обеспечить качество многоязычных приложений компьютерной обработки текстов.

Прогресс в этой области связан с решением двух промежуточных задач: автоматическое разрешение лексической многозначности и масштабное обогащение лексических баз знаний.

Проблема, однако, заключается в том, что существуют взаимозависимые факторы:

1. для того, чтобы достичь качественного разрешения лексической многозначности, необходимо значительно больше лингвистического и семантического знания, чем имеется в текущих лексических базах знаний (к примеру, в ворднетах)
2. для того, чтобы обогатить существующие лексические базы знаний необходимо получать информацию из корпусов с качественной семантической разметкой.

В проекте планируется выполнить три последовательных цикла масштабного разрешения лексической многозначности и извлечения знаний для пяти европейских языков, включая баскский, испанский, итальянский, голландский и английский языки. Накопленные знания должны храниться в Многоязычном Центральном Репозитории.

Эксперименты по семантическому индексированию в рамках проекта Meaning

В рамках европейского проекта Meaning голландская компания Irion Technologies разработала технологию концептуального индексирования TwentyOne, комбинирующую лингвистический и статистический подходы (Vossen 2006). Авторы разработки считают, что неудачи с использованием WordNet в информационно-поисковых приложениях связаны с трудностями

встраивания такого рода лингвистических ресурсов в приложения, оптимального использования содержащейся в ворднетах информации.

Основой технологии является статистическая машина поиска, базирующаяся на стандартной векторной модели и которая обеспечивает быстрый поиск документов.

Лингвистические технологии используются в 2 ролях:

- максимизация полноты выдачи статистической машины за счет синонимии ворднетов
- максимизация точности выдачи за счет сравнения запросов с конкретными фразами документов, а не с целыми документами.

Фраза представляет собой именную группу (noun phrase). Каждая фраза ассоциируется с отдельными словами, определенной комбинацией слов, а также комбинацией частей слов

Система TwentyOne использует совокупность факторов для сравнения запроса с фразами текста:

- Число совпадающих концептов между запросом и каждой фразой,
- Степень нечеткого сопоставления между запросом и каждой фразой,
- Степень деривационного несовпадения, слитного – раздельного написания и т.п.
- были ли использованы синонимы
- был ли использован тот же язык

Суть технологии в том, что сначала выдаются документы, которые имеют наибольшее совпадение по концептам фраз с запросом. Среди документов, имеющих одинаковое количество сопоставленных понятий между собственными фразами и запросом, первыми выдаются наиболее похожие по конкретному набору слов.

В проводимых экспериментах для сравнения были построены четыре индекса:

1. HTM – традиционный пословный индекс
2. NP - индексы именных групп из запроса, с использованием пословных методов, без использования ворднетов
3. FULL - полные индексы с использованием ворднетов, но без процедуры разрешения многозначности, что приводит к полному расширению по синонимам и переводам для всех возможных значений слов запроса

4. WSD - индексы, использующие ворднеты вместе с описанной выше процедурой снижения многозначности на основе предметных областей ворднет.

В эксперименте индексы тестируются в системе автоматической рубрикации текстов на коллекции Reuter. Описывается, что максимальных значений F-меры система автоматической рубрикации достигает для индекса WSD: полнота – 80.7, точность 72.2. Минимум система имеет на базе индекса HTM: полнота – 67.8, точность 70.4. Нужно однако отметить, что описываемые результаты на основе пословного индекса значительно ниже, чем результаты других пословных систем на этой же коллекции. Получается, что произведенные улучшения получаются по сравнению с заниженным недостаточно эффективным уровнем работы системы на основе пословного индекса, и значительные улучшения могли бы быть осуществлены еще в рамках такого индекса.

Вопросы к лекции

1. Что такое концептуальное индексирование и концептуальный поиск?
2. Какие проблемы использования онтологии в информационном поиске?

Литература

1. *М.С. Агеев, Б.В. Добров, Н.В.Лукашевич, А.В. Сидоров, С.В. Штернов* . Отправная точка для дорожки по поиску в РОМИП. – РОМИП 2003.
2. Salton G., Automatic Text Processing - The Analysis, Transformation and Retrieval of Information by Computer. Addison-Wesley, Reading, MA, 1989
3. Voorhees, E. (1993) Using WordNet to Disambiguate Word Senses for Text Retrieval , in Proc. 16th Annual ACM SIGIR Conference on Research and Development in ...
4. Voorhees E., Natural Language Processing and Information Retrieval. // M.T. Pazienza, (Ed.), Information Extraction: Towards Scalable, Adaptable Systems, Germany: Springer, 1999, pp.32-48.
5. Vossen, Piek, Rigau G., Alegria I., Agirre E., Farwell D., Fuentes M. Meaningful results for Information Retrieval in the MEANING project. Proceedings of Third International WordNet Conference.
6. Woods W.A., Conceptual indexing: a better way to organize knowledge. *SunMicrosystems Laboratories Technical Report*, SMLI TR-97-61. - 1997.

8.3. WordNet: Применение в вопросно-ответных системах

Вопросно-ответная система представляет собой вид информационно-поисковых систем. Вопросно-ответная система должна предоставить не набор документов, которые наиболее релевантны поставленному вопросу, но выдать точный ответ на данный вопрос.

Разработки вопросно-ответных систем были начаты в 60-е годы. В то время предполагалась, что ответ на вопрос должен искаться в специально подготовленных базах знаний.

Второе рождение вопросно-ответные системы начали переживать с 90-х годов 20 века. Теперь вопросно-ответные системы должны искать ответы в больших текстовых коллекциях.

С 1999 года проводится соревнование по вопросно-ответным системам в рамках конференции TREC (Voorhees 2002), с 2003 года соревнования вопросно-ответных систем в многоязычном контексте начаты на конференции CLEF (Magnini et.al. 2005).

Соревнование систем в рамках этих конференций проводится следующим образом:

Участникам рассылается большой текстовый массив (более нескольких гигабайт) и порядка 200 вопросов. Нужно прислать текстовые фрагменты (50, 250 байт), содержащие ответы на вопрос. Ответы должны быть упорядочены, засчитываются первые три ответа.

Оценка производится следующим образом: за первый правильный ответ система получает 1 балл, за правильный ответ на втором месте – 0.5 балла, на третьем месте – 0.25.

Общая оценка системы получается вычислением среднего балла по всем вопросам.

Основными этапами поиска ответа на вопрос в современных вопросно-ответных системах являются следующие:

Прежде всего, производится подробный анализ вопроса, в результате которого определяется тип вопроса (вопрос времени, места, количества и другие) и соответствующий тип ответа, а также формируется запрос к информационно-поисковый системе.

На втором этапе производится поиск релевантных документов или абзацев информационно-поисковой системой, формируется упорядоченный список наиболее релевантных документов (абзацев), из которого выбирается первых n (например, n=100) документов (абзацев) для дальнейшей обработки.

На третьем этапе производится подробный анализ полученных абзацев: содержит ли абзац требуемые тип ответа, близость слов ответа и вопроса и т.п.

Вопросы как особый тип запросов к информационно-поисковой системе

Как известно, запросы в глобальных информационно-поисковых системах обычно очень короткие 2-3 слова, и по ним находятся сотни и тысячи документов.

Запросы в форме вопросов обычно значительно длиннее. При этом в многих современных исследованиях по вопросно-ответным системам пользуются не векторными моделями поиска, а выполняют булевский поиск, поскольку считается, что при выполнении данной задачи необходимо осуществлять контроль, какие слова формулировки вопроса обязательно должны присутствовать в тексте ответа.

Булевское выражение обычно формируется как конъюнкция всех значимых слов формулировки вопроса. Если проводится морфологический анализа запроса или добавляются синонимы, то они объединяются в дизъюнкцию.

Например, если задан вопрос *When did Shapour Bakhtiar die?,* то может быть образовано следующее булевское выражение:

Shapour *AND* ***Bakhtiar***
AND ***(die*** *OR* ***dies*** *OR* ***died*** *OR* ***dying*** *OR* ***died*** *OR* ***death****)*

Однако стандартной является ситуация, когда не находится документов, которые содержат все значимые слова вопроса, поэтому при обработке вопроса часто необходимо определить, какие именно слова формулировки вопроса можно отбросить, не включить в поисковый запрос без потери сути вопроса.

Например, следующему вопросу *«Кто из великих целителей прошлого написал трактат "О медицине“?»* может частично соответствовать два предложения:

1. *ЦЕЛЬС (Celsus) Авл Корнелий (I в. до н. э.), древнеримский автор энциклопедических трудов «Artes» (сохранился* ***трактат "О медицине",*** *книги 1 - 8, с ценными сведениями по гигиене, хирургии, дерматологии)*
2. *А.Е. Ферсман приводит отрывок из* ***трактата*** *"Сокровищница лекарств",* ***написанного*** *арабским* ***целителем*** *около тысячи лет назад: "Ношение бирюзы…*

Первое из предложений содержит правильный ответ *ЦЕЛЬС,* во втором предложении кандидатом на ответ является *А.Е.Ферсман*, что неверно.

Таким образом, часто сформулированный по формулировке вопроса булевский запрос к информационно-поисковой системе не находит ни одного документа. Поэтому обычно предлагается система модификаций, упрощающих исходное булевское выражение, после каждой из которой опять происходит обращение к поисковой системе для проверки, не появились ли релевантные документы.

Используются обычно два основных способа упрощения булевского выражения.

Во-первых, можно часть конъюнкций переводить в дизъюнкции.

Вторым способом является поочередное исключение членов конъюнкции, на основе некоторого множества эвристик, определяющих значимость членов конъюнкции [Harabagiu et.al. 2001].

Значимость членов конъюнкции может определяться на основе их грамматических характеристик в формулировке вопроса. Так, наиболее значимыми обычно считаются имена, фразы в кавычках, а наименее значимыми считаются глаголы [Moldovan et.al 1999].

Процесс исключения элементов из конъюнкции прекращается, когда количество документов (абзацев) в выдаче достигает заданного числа (например, 50) или до тех пор пока не остается заданный процент слов исходной формулировки вопроса [Magnini et.al, 2003].

В связи с длинной формулировкой естественно-языкового вопроса и частым отсутствием в самых больших текстовых коллекциях ответов, содержащих все или большинство слов формулировки вопроса, значимой становится роль лексических ресурсов, позволяющих найти ответы в тех предложениях, в которых часть слов заменена на близкие по смыслу слова.

Так, например, ответ на вопрос Почему электрические батареи быстрее разряжаются на холоде? может быть следующим: *Батарейки быстрее садятся на морозе, потому что..».*

Таким образом, практически каждое слово вопроса имеет соответствующее слово в данном ответе, при этом сделано 3 лексические замены. Пример не придуман, а выявлен в процессе одного из проводимых автором экспериментов. Более лексически точного ответа в текстовой коллекции не нашлось.

WordNet в вопросно-ответной системе Южного Методистского университета США

Одной из самых эффективных систем в вопросно-ответной дорожке конференции TREC стала вопросно-ответная система Южного Методистского университета, которая на нескольких этапах обработки вопроса и поиска ответа обращается к информации, хранимой в WordNet. В разработанной системе WordNet используется для:

- распознавания типа вопроса;
- классификации типов ответов;
- для реализации лексических и семантических замен.

Лексические и семантические замены осуществляются в момент сопоставления формальной структуры вопроса и ответа. Поиск в системе организован на основе обработки булевских запросов.

Реальные вопросы

Стоит отметить, что вопросы, на которые могла бы искать ответ вопросно-ответная система, очень востребованы в таких случаях, когда, например, люди обращаются в какие-либо компьютерные форумы с просьбой помочь им в решении какой-либо проблемы (например, с поломкой компьютера или в какой-либо правовой ситуации). В таких случаях часто оказывается, что такие ситуации уже обсуждались и достаточно просто найти соответствующие ответы, которые однако были сформулированы несколько иначе, поэтому в простом пословном поиске найти предшествующие аналогии очень трудно.

Однако, такие просьбы о помощи редко формулируются как простой, правильно построенный вопрос. Чаще, они включают несколько предложений с описанием проблемы и, возможно, более одного вопроса. Например, вопрос может выглядеть таким образом:

> *Ноутбук Compaq nx9010, месяц от роду, лицензионная русская XP Home SP1, каждые 3-4 дня загадочно исчезают точки восстановления: просто стираются соответствующие папки. Похоже, что при перезагрузке. Но не уверен. В календаре мастера восстановления - тоже исчезают. На диске свободно 27 Гб, движок стоит на все 12%. На десктопе со времён установки XP ничего подобного никогда не наблюдалось (там без сервиспака). Принятые меры: выключение и снова включение восстановления - ноль внимания. Снесение*

системы, установка заново – аналогично. Где копать? Машина хорошая, претензий нет. К виндам во всём остальном – тоже. Железо? Винды? Хитрые дрова? Что?

Вопросы к лекции

1. Основные этапы работы вопросно-ответной системы
2. Как можно использовать онтологию в вопросно-ответной системе
3. Обработка булевского запроса в вопросно-ответной системе

Литература

1. Harabagiu S., Moldovan D., Pasca M., Mihalcea R., Surdeanu M., Bunescu R., Girju R., Rus V., Morrescu P. (2001) The role of lexico-semantic feedback in open-domain textual question-answering. In Proceedings of the Association for Computational Linguistics, pages 274281, July 2001.
2. Harabagiu S., Moldovan D., Clark C., Bowden M., Williams J., Bensley J. (2004) Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In: Voorhees E.M. and Buckland L.P. (eds) Proceedings of the Twelfth Text Retrieval Conference (TREC 2003), 375-382.
3. Marius Pasca and Sanda Harabagiu. The Informative Role of WordNet in Open-Domain Question Answering. (NAACL 2001). (dingo.sbs.arizona.edu/~sandiway/csc620/eggers.pdf)
4. Moldovan D., Harabagiu S., Pasca M., Mihalcea R., Goodrum R., Girji R., Rus V.(1999). LASSO: A tool for surfing the answer net. In Proceedings of the Eighth Text Retrieval Conference (TREC-8)
5. Magnini B., Negri M., Prevete R., Tanev H. Multilingual Question/Answering: the DIOGENE System. In Proceeding of the TREC-10 Conference, pages 335--344, Gaithesburg, MD, 2001.
6. Soubbotin, M. "Patterns of Potential Answer Expressions as Clues to the Right Answers" in *Proceedings of the 10th Text Retrieval Conference,* pp. 293-302, NIST, Gaithersburg, MD, 2002.
7. Soubbotin, M. and Soubbotin, S. "Use of Patterns for Detection of Answer Strings: A Systematic Approach" in *Proceedings of the 11th Text Retrieval Conference,* pp. 325-331, NIST, Gaithersburg, MD, 2003.

8.4. WordNet. Проблемы использования в автоматической обработке

Многозначность в WordNet

Во многих работах признается, что различия значений в WordNet слишком тонки для таких компьютерных приложений как машинный перевод, информационный поиск, классификация текстов, вопросно-ответные системы и др. В (Chugur 2002) было показано, что среднее количество значений в WordNet больше, чем в традиционных лексикографических словарях.

Проблема лексической многозначности и информационный поиск

В (Chugur et.al. 2000) исследуется вопрос, какая группировка значений была бы полезной для задач информационного поиска. Предполагается, что некоторые значения могут быть кластеризованы для разных приложений, в то же время существуют примеры пар значений, кластеризация которых была бы полезна в информационно-поисковых приложений, при этом в других приложениях было бы полезно их различать.

Отмечается, что исследования регулярной многозначности не приводят к выделению полезных кластеров для информационно-поисковых задач, так как, как представляется авторам, некоторые образцы регулярной полисемии хорошо бы не различать для задач информационного поиска, в то время как другие хорошо бы сохранить отдельно. Так, например, полезно было бы кластеризовать такие пары регулярной полисемии как *container/quantity и music/dance*. Однако такие образцы как *animal/food, plant/food, animal/skin, language/people* хорошо бы различать, поскольку, как представляется они употребляются в разных типах текстов.

Поэтому нужны дополнительные исследования критериев кластеризации значений для информационно-поисковых задач.

В работе сравниваются два дополнительных критерия группировки значений. Первый критерий заключается в том, чтобы группировать значения, которые встречаются в одних и тех же текстах. Для этого используется семантически размеченный корпус Semcor. Второй критерий группирует значения, которые получают одни и те же переводы в нескольких разных языках. Пересечение кластеров, построенных на основе этих двух критериев составляет 55-60 процентов, что показывает некоторую корреляцию между кластерами, но оставляет сомнения в полезности каждого из критериев.

Проведенные эксперименты по кластеризации значений привели авторов к выводу, что типология отношения между разными значениями

многозначных слов является более полезной, чем формирование кластеров значений, поскольку близость значений зависит от приложения.

Например, указание, что одно из значений является метафорой исходного значения является важным различие для приложений информационного поиска и вопросно-ответных систем, поскольку относится к разным семантическим полям. Однако для приложений машинного перевода это различие может быть несущественно, поскольку метафорический перенос может быть сходным в разных языках. В ворднетах нужно эксплицитно описать отношения между значениями для того, чтобы ворднеты стали стандартом лексических ресурсов для компьютерных приложений.

Tennis problem

[illegible] WordNet, [illegible] «[illegible]»: принадлежащие одной предметной области, сфере деятельности, ситуации синсеты, оказываются очень далеко друг от друга в структуре WordNet.

Предлагалось данную проблему решать введением в WordNet информации о принадлежности синсетов определенным доменам. Домены такие как «теннис», «политика» или «образование» группируют синсеты в сценарии или схемы. Так, такой домен как «теннис» включает такие синсеты как «гейм», «теннисный мяч», «теннисная ракетка», «тай-брейк» и т.д.

Предполагается, что введение доменов должно быть особенно полезно для информационно-поисковых задач.

Работа [Magnini.. 2000] описывает процесс создания иерархической системы таких доменов и процедуру автоматизированной приписки доменов синсетам WordNet.

Разработка иерархической системы доменов началась с 250 рубрик, собранных по различным словарям и затем была дополнена и уточнена на базе Десятичной классификации Дьюки. Была получена иерархия из 115 доменов, организованных по 4 уровням иерархии, включающая, например, такие домены как например, такие домены как «сельское хозяйство», «археология», «астрология», «биология», «ветеринария» и др..

Авторы подчеркивают, что в ворднет имеются синсеты, которые не принадлежат никаким доменам, поскольку они могут употребляться в текстах многих предметных областей. Для таких синсетов была введена специальная предметная область, называемая FACTOTUM.

Для того, чтобы разметить все множество синсетов WordNet была реализована автоматизированная процедура, состоящая из следующих шагов:

1. Вручную размечается относительно небольшое количество синсетов верхнего уровня;
2. Автоматически по связям (гипонимия, тропонимия, меронимия, антонимия) пометки распространяются на другие синсеты;
3. Можно задать исключения, например, для синсета кресло парикмахера («barber_chair»), которое является частью парикмахерской («barbershop») и поэтому получает домен КОММЕРЦИЯ (COMMERCE).

Эксперименты с доменами в ворднетах были продолжены и в следующем европейском проекте, связанном с ворднетами, Meaning (Atserias et.al., 2004), (Castillo et.al 2004), в котором было 165 иерархически организованных доменов были автоматизировано приписаны всем синсетам WordNet.

Авторы также подчеркивают полезность разметки синсетами доменами для автоматического разрешения лексической многозначности.

Вместе с тем остаются вопросы по отношению к введению в систему, построенную на основе одних единиц, набора других единиц с неопределенным относительно исходных единиц статусом среди которых:

- Вариативность возможного набора областей,
- небольшая наполненность некоторых доменов, и большое количество синсетов в других доменах,
- необходимость разных систем доменов для разных задач,
- отсутствие полностью выверенного набора доменов (выверить вручную очень трудоемко, если выверять в процессе решения различных задач, то далеко не все проблемы (неточности, ошибки) приписки удастся быстро обнаружить).

Представление толкований WordNet виде логических выражений: проект в eXtended WordNet (Rada Michalchea and Dan Moldovan)

Многие исследователи отмечают нехватку информации, описанной в WordNet, значений, в нем перечисленным.

В рамках проекта eXtended WordNet разработчики предполагают, что важным источником дополнительной информации могут стать толкования, приписанные к синсетам WordNet. Для того, чтобы эти толкования можно было использовать в автоматических режимах компьютерных приложений необходимо каждому знаменательному слову толкований сопоставить его

значение-синсет и представить это толкование в виде формализованного выражения.

Разрешение лексической многозначности (Word Sense Disambiguation)

Поскольку стало ясно, что применению таких ресурсов как WordNet препятствует такая проблема, как недостаточная эффективность разрешения лексической многозначности, то эта проблема получила отдельную значимость.

Была организована специальная конференция, посвященная проблеме разрешения лексической многозначности - Конференция SENSEVAL.

Первая конференция по оценке методов разрешения лексической многозначности SENSEVAL состоялась в 1998 году, охватывала три языка, в ней приняли участие 25 исследовательских групп. Вторая конференция состоялась в 2001 году, имела задания на 12 языках; участвовали 35 исследовательских групп и более 90 систем.

Вопросы к лекции

1. Каковы проблемы, возникают при использовании WordNet для автоматической обработки текста?
2. Опишите проблему лексической многозначности.
3. Как в WordNet происходит разрешение многозначности?

Литература

1. Кобрицов Б.П. Методы снятия семантической многозначности // Научно-техническая информация, сер.2, 2004а, N 2.
2. Рахилина Е.В., Кобрицов Б.П., Кустова Г.И., Ляшевская О.Н., Шевинаева О.Ю. Многозначность как прикладная проблема: лексико-семантическая разметка в национальном корпусе русского языка. Диалог 2006, стр. 4450 - 450.
3. Agirre E., Lacalle Lopez O. Clustering Wordnet word senses. In proceedings RANLP 2003.
4. Castillo M., Real F., Rigau G. Automatic Assignment of Domain Labels to WordNet. - *In Proceedings of International Wordnet Conference (-GWC – 2004)*. – 2004. – pp. 75-82.
5. Chugur I., Gonzalo J., Verdejo F. A study of sense clustering criteria for information retrieval applications. In proceedings of OntoLex 2000.
6. Chugur I., Gonzalo J., Verdejo F. Polysemy and sense proximity in the Senseval-2 Test Suite. In Proceedings of the ACL-2002 Workshop on “Word sense Disambiguation: recent successes and future directions”, 2002.

7. Gonzalo J. Chugur I., Verdejo F. Sense clustering for information retrieval: evidence from Semcor and the EWN Interlingual Index. In the Proceedings of the ACL 2000 Workshop on Word Senses and Multilinguality.
8. Kilgariff A., Rosenzweig J. Framework and results for English SENSEVAL. Computers and Humanities, 34, 2000, p.15-48.
9. Magnini B., Cavaglia G. Integrating Subject Field Codes into WordNet. – In proceeding of the Second International Conference on Language Resources and Evaluation LREC 2000, Athens, Greece.
10. Mihalcea R. Moldovan D.I. (2001) Automatic generation of a coarse grained WordNet. In Proceedings of SIGLEX
11. Mihalcea R. Moldovan D.I . eXtended WordNet: progress report. In NAACL 2001

9. ТЕЗАУРУСЫ. ОСНОВНЫЕ ПРИНЦИПЫ РАЗРАБОТКИ, СОЗДАНИЯ И ИСПОЛЬЗОВАНИЯ ТРАДИЦИОННЫХ ИНФОРМАЦИОННО-ПОИСКОВЫХ ТЕЗАУРУСОВ. ПРИМЕРЫ ТЕЗАУРУСОВ.

Начало разработки информационно-поисковых тезаурусов для различных предметных областей относится к середине 60-х годов. В то время большинство информационных систем не являлись полнотекстовыми, а хранили достаточно ограниченный набор информации о документе: библиографические данные, реферат. Добавление списка ключевых слов, характеризующих основное содержание документа, существенно расширяли возможности поиска документов. С начала семидесятых годов создаются национальные и международные стандарты разработки информационно-поисковых тезаурусов

С появлением полнотекстовых информационно-поисковых систем, а также возможностей поиска по всем словам текста с помощью методов ранжированного информационного поиска значительно снизило значимость разработки и использования информационно-поисковых тезаурусов, поскольку давало возможность поиска текста неподготовленному пользователю в любых предметных областях, без предварительных затрат на разработку тезаурусов.

Потенциально использование тезаурусов в качестве средств для описания основного содержания текста позволяет преодолевать многие проблемы пословного поиска такие как:

- избыточность -- в пословном индексе используются слова-синонимы, выражающие одни и те же понятия;
- слова текста считаются независимыми друг от друга, что не соответствует свойствам связного текста;
- многозначность слов --поскольку многозначные слова могут рассматриваться как дизъюнкция двух или более понятий, выражающих различные значения многозначного слова, то маловероятно что все элементы этой дизъюнкции интересуют пользователя;
- приписанных текстов слов так много, что возникает отдельная проблема по определению их значимости для данного текста и другие.

Однако многочисленные исследования по определению эффективности различных методов представления документов при информационном поиске показали, что эффективность пословного индексирования сравнима с

эффективностью поиска, использующего ручное индексирование по тезаурусу.

Действительно, использование хорошо разработанного тезауруса при ручном индексировании должно снимать проблемы синонимии, близких понятий, многозначности. Однако при этом могут возникнуть существенные различия между понятиями, используемыми в тезаурусе, и информационной потребностью пользователя, когда пользователю трудно сформулировать описание нужных ему текстов посредством понятий тезауруса, или тезаурус действительно не содержит адекватных понятий. В этих случаях пословное индексирование имеет преимущество из-за больших выразительных возможностей.

Кроме того, при ручном индексировании серьезную проблему составляет фактор субъективности, когда приписывание тексту терминов тезауруса зависит от умения и опыта индексаторов, от количества текстов, которые необходимо проиндексировать и т.п.

Тем не менее и в настоящее время существуют информационные службы, имеющие и разрабатывающие информационно-поисковые тезаурусы, а также имеющие штат профессиональных индексаторов, индексирующих документы на основе тезаурусов. Примерами таких организаций являются Исследовательская служба Конгресса США, индексирующая по тезаурусу Legislative Indexing Vocabulary, Продовольственная и Сельскохозяйственная организация при ООН (ФАО) развивает тезаурус AGROVOC, службы Европейского сообщества используют для индексирования Европейского законодательства тезаурус EUROVOC и др. Происходит и процесс обновления стандартов разработки тезаурусов.

За прошедшие годы были разработаны и использовались информационными и терминологическими службами сотни тезаурусов, каждый из которых содержит ценную информацию о своей предметной области. Поэтому многие разработчики автоматических информационных систем исследовали вопросы о применении существующих информационно-поисковых тезаурусов при обработке документов в автоматическом режиме. Однако подавляющее большинство экспериментов окончились неудачей: применение информационно-поисковых тезаурусов в процессе автоматического индексирования увеличивало полноту поиска, но резко снижало его точность.

Более того, международный стандарт по разработке одноязычных тезаурусов (ISO 2788) четко указывает, что стандарт должен применяться в организациях, имеющих людей-индексаторов, которые анализируют

содержание документов и описывают основные темы документов с помощью терминов тезауруса. «Применение стандарта не предполагает его применение в тех организациях, которые используются полностью автоматические методы индексирования».

Возникает вопрос, почему существующая парадигма разработки информационно-поисковых тезаурусов не дает возможности использовать созданные ресурсы в автоматических режимах индексирования текста. Как и можно ли создавать тезаурусы для автоматического индексирования? Для этого необходимо разобраться, какие особенности существующей парадигмы разработки информационно-поисковых тезаурусов не позволяют их использование в автоматических режимах. В дальнейшем тексте информационно-поисковые тезаурусы, создаваемые в соответствии с существующими международными и национальными стандартами, будем называть традиционными информационно-поисковыми тезаурусами.

Для этого рассмотрим структурные особенности традиционных информационно-поисковых тезаурусов.

Назначение информационно-поисковых тезаурусов

В различных стандартах и пособиях приводятся разные определения информационно-поисковых тезаурусов.

Объемлющее определение информационно-поискового тезауруса можно сформулировать следующим образом: Информационно-поисковый тезаурус – это контролируемый словарь терминов на естественном языке, явно указывающий отношения между терминами и предназначенный для информационного поиска.

Основными целями разработки традиционных информационно-поисковых тезаурусов являются следующие:

- обеспечение перевода естественного языка документов и пользователей на контролируемый словарь, используемый для индексирования и поиска
- обеспечение последовательного использования единиц индексирования,
- обеспечение отношений между терминами,
- использование как поисковое средство при поиске документов.

Единицы традиционных информационно-поисковых тезаурусов

Основной единицей тезаурусов являются термины, которые разделяются на дескрипторы (=авторизованные термины) и недескрипторы (=аскрипторы).

Большинство версий стандартов по информационно-поисковым тезаурусам указывают на связь терминов с понятиями предметной области.

Американский стандарт указывает, что термин является одним или большим числом слов, обозначающих понятие. Стандарт ISO подчеркивает, что индексирующий термин - это представление понятия предпочтительно в форме существительного или именной группы.

При этом понятие рассматривается как единица мысли, формируемая мысленно для отражения всех или некоторых свойств конкретного или абстрактного, реально существующего или мысленного объекта. Понятия существуют как абстрактные сущности, независимо от терминов, которые их выражают.

Российский ГОСТ рассматривает понятие как форму мышления, отражающую существенные свойства, связи и отношения предметов и явлений, а термином в определении ГОСТа является слово или словосочетание, являющееся точным обозначением определенного понятия какой-либо области знания.

При этом, определяя единицы тезауруса ГОСТ7.74-96 не опирается на определение термина, а определяет единицы тезауруса как лексические единицы информационно-поискового языка – то есть обозначения отдельного понятия, принятые в информационно-поисковом языке и неделимое в этой функции.

Стоит отметить, что не все разработчики тезаурусов четко разделяли понятия и термины. Так, разработчики тезауруса AGROVOC характеризуют его как термино-ориентированный (term-oriented), что находит свое проявление в том, что к термину не возможно добавить синонимы. Эта особенность тезауруса рассматривается авторами как недостаток, который необходимо исправить (Soergel и др. 2004).

Таким образом, разработчики тезаурусов предполагают, что понятие предметной области обычно имеет несколько возможных вариантов лексического представления в тексте, которые рассматриваются как синонимы. Среди таких синонимов выбирается дескриптор – термин, который рассматривается как основной способ ссылки на понятие в рамках тезауруса. Другие термины из синонимического ряда, включенные в тезаурус, называются аскрипторы или недескрипторы. Они используются как вспомогательные элементы, текстовые входы, помогающие найти подходящие дескрипторы.

Дескрипторы

Дескрипторы тезауруса должны соответствовать выбранной предметной области тезауруса. Каждый дескриптор, внесенный в тезаурус, должен представлять отдельное понятие данной области. Дескриптор может быть однословным или многословным. Поскольку часто достаточно трудно понять,

представляет ли отдельное понятие многословное словосочетание, многие тезаурусы и руководства уделяют особое внимание основным принципам включения в тезаурус в качестве дескрипторов многословным терминов

Набор дескрипторов должен удовлетворять следующим требованиям:

- посредством выделенных дескрипторов должно быть возможно описать темы абсолютного большинства текстов предметной области;
- для уменьшения субъективности индексирования множество дескрипторов не должно включать совокупности близких дескрипторов, формируются классы условной эквивалентности, когда совокупности близких, но различных понятий сводятся к одному дескриптору (LIV, 1994);
- дескриптор должен быть сформулирован однозначно, его подразумеваемое в рамках тезауруса значение должно быть понятно пользователю. Если однозначный и ясный дескриптор подобрать не удается, термин, взятый в качестве дескриптора снабжается релятором (краткой пометой) или комментарием.

Отношения в информационно-поисковом тезаурусе

Большинство информационно-поисковых тезаурусов включают два основных отношения между дескрипторами:

- отношения ВЫШЕ-НИЖЕ (Broader Term – Narrower term)
- отношение Ассоциации (Related Term).

Ассоциативные отношения

Основным назначением установления ассоциативных отношений между дескрипторами информационно-поискового тезауруса является то, что установление такой связи может указать дополнительные дескрипторы, полезные при индексировании или поиске.

Отношение ассоциации является неиерархическим и ассоциативным. Ассоциативное отношение наиболее трудно определить. Российский стандарт на создание информационно-поисковых тезаурусов указывает, что «ассоциативное отношение является объединением отношений, не входящих в иерархические отношения или в отношения синонимии. Допускается включать в ассоциативное отношение все виды отношений, кроме синонимии и отношения род — вид.» (СИБИД 7.25-2001).

Другие источники стараются изложить более подробные принципы установления ассоциативных отношений, поскольку в противном случае отношение будет проставляться непоследовательно.

Американский стандарт описывает наиболее общее правило установления ассоциативного отношения между дескрипторами таким

образом, что это отношение стоит устанавливать между двумя дескрипторами, если при употреблении одного термина другой термин как бы подразумевается. Более того, один термин часто необходимый элемент определения другого термина, например, термин *клетка* составляет необходимую часть определения термина *цитология*.

Автоматическое индексирование по традиционным информационно-поисковым тезаурусам

В работе (Hlava, Heinebach, 1996) излагается подход к автоматическому индексированию по тезаурусу EUROVOC, основанному на правилах. Правила могут быть простыми и сложными. Простые правила не содержат условий. Сложные правила содержат такие условия как Близость (на расстоянии трех слов по тексту, в одном предложении, в том же самом поле, например, поле реферата), Местонахождение (в заголовке, в тексте реферата или документа, начало предложения, конец предложения), Формат (с большой буквы, все большими буквами). Всего было создано около 40 тысяч правил.

В работе (Steinberger и др., 2000) автоматическое приписывание дескрипторов тезауруса EUROVOC полнотекстовым документам включает две стадии.

На первой стадии (этап обучения) на основе документов, в ручную проиндексированных индексаторами, устанавливается соответствие между словами, встретившимися в тексте документа, и приписанными дескрипторами тезауруса. Соответствие устанавливается на основе статистических мер (chi-square или log-likelihood). Вес соответствия отдельного слова ключевому слову тем выше, чем выше совместная частотность использования данного слова и данного ключевого слова относительно частотности во всей коллекции.

Например, дескриптору тезауруса FISHERY MANAGEMENT соответствуют следующие слова (в порядке убывания веса): fishery, fish, stock, fishing, conservation, management, vessel, и т.д.

На второй стадии (собственно, индексирование) для каждого слова документа проверяется, каким дескрипторам тезауруса оно соответствует. Если такие дескрипторы имеются, то слово добавляет к весу дескриптора для данного текста натуральный логарифм веса, полученного на первом этапе. После обработки всех слов текущего текста получается суммированный вес дескрипторов тезауруса.

Например, для Резолюции по правам языковых и культурных меньшинств в Европейском союзе были получены следующие дескрипторы (в

порядке убывания веса). *Community programme, Young person, cultural policy, CEEC, European Union и т.д.*

Индексаторы Европейского Парламента присваивают документу обычно от 3 до 10 дескрипторов.

Выдачу системы можно ограничить по количеству выдаваемых дескрипторов или по весу. Для текста примера присвоенные индексаторами дескрипторы находились в первой тридцатке дескрипторов, присвоенных автоматически (на позициях 3, 8, 9, 16 и 30).

При этом авторы подчеркивают, что большинство автоматически присвоенных дескрипторов выглядят весьма релевантными тексту документа, и только 3 из 40 присвоенных автоматически явно неправильны (например, Кипр).

Сочетание свободных запросов и запросов на основе информационно-поисковых тезаурусов

В настоящее время в мире существует достаточно много информационных систем, предоставляющих пользователям как возможности поиска информации по свободному запросу на естественном языке, так и с помощью дескрипторов информационно-поисковых тезаурусов, сопоставленных документам профессиональным индексаторами.

Первым шагом на таком пути может быть нахождение корреляций между словами документов и дескрипторами тезауруса или рубриками рубрикатора.

Эксперименты по автоматическому расширению свободного запроса пользователя дескрипторами тезауруса описаны в работах (Petras 2004; Petras 2005).

Эксперименты проводились на двуязычной коллекции немецких и английских документов по общественным наукам. База содержит более 150 тысяч немецких документов и 26 тысяч английских документов. Документы реферативного характера содержат заголовок публикации, реферат и дескрипторы Тезауруса по общественным наукам, приписанных индексаторами. Эксперименты выполнялись в рамках предметно-ориентированного задания форума по многоязыковым информационным система CLEF.

Вопросы к лекции

1. Перечислите основные виды отношений в ИПТ.
2. Почему традиционные ИПТ мало используются для автоматического индексирования текстов.

3. Методы использования традиционных ИПТ в автоматических технологиях обработки текстов (запросов).

Литература

1. Архангельская В.А., Базарнова С.В. Информационно-поисковый тезаурус по экономике и демографии. – НТИ, сер.1. Орг. и методика информ. работы. – 2001, N 7, стр. 24-32.
2. Жмайло С.В. К разработке современных информационно-поисковых тезаурусов. – НТИ, сер.1, 2004, N1, стр. 23-31.
3. Мдивани Р.Р. О разработке серии тезаурусов по социальным и гуманитарным наукам. – НТИ, сер.2. Информ. процессы и системы. 2004. - N 7, стр. 1-9.
4. СИБИД. Тезаурус информационно-поисковый одноязычный: Правила разработки: структура, состав и форма представления: Межгосударственный стандарт 7.25. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2001.
5. Список нормализованной лексики по экономике и демографии. - М.: АН СССР, ИНИОН, 1989.- Ч. 1. - 169 с. 1.
6. Черный А.И. Общая методика построения тезаурусов. – НТИ. Сер.2. – 1968. – N5. – С9-32.
7. Шемакин Ю.И. Тезаурус в автоматизированных системах управления и информации. - М: Военное изд-во министерства обороны СССР, 1974. - 192 с.
8. Шемакин Ю.И. Тезаурус научно-технических терминов. - М: Военное изд-во министерства обороны СССР, 1972.
9. Hlava M., Hainebach R. Multilingual Machine Indexing. - Proceedings of The Ninth International Conference on New Information Technology, Pretoria, South Africa, November 11-14, 1996. – pp. 105-120.
10. LIV (Legislative Indexing Vocabulary). Congressional Research Service. The Library of Congress. Twenty-first Edition. – 1994.
11. Petras V. GIRT and the Use of Subject Metadata for Retrieval. In: Multilingual Information Access for Text, Speech and Images. 5th workshop of the Cross-language Evaluation Forum, CLEF 2004. Lecture Notes in Computer Science, Vol. 3491. Springer-Verlag. 2004.– pp. 298-309.
12. Petras V. How One Word Can Make all the Difference – Using Subject Metadata for Automatic Query Expansion and Reformulation. - In: Multilingual Information Access for Text, Speech and Images. 6th workshop of the Cross-language Evaluation Forum, CLEF 2005. Lecture Notes in Computer Science, Springer-Verlag. 2005.

13. Steinberger R., Hagman J. Scheer St. Using Thesauri for Automatic Indexing and Visualisation. – In Proceedings OntoLex 2000. – p. 130-141.
14. Z39.19 – Guidelines for the Construction, Format and Management of Monolingual Thesauri. – NISO, 1993.

10. ИНФОРМАЦИОННО-ПОИСКОВЫЕ ТЕЗАУРУСЫ В УСЛОВИЯХ СВЕРХБОЛЬШИХ ЭЛЕКТРОННЫХ КОЛЛЕКЦИЙ И АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТОВ. ТЕЗАУРУС ДЛЯ АВТОМАТИЧЕСКОГО КОНЦЕПТУАЛЬНОГО ИНДЕКСИРОВАНИЯ КАК ОСОБЫЙ ВИД ТЕЗАУРУСА

10.1. Тезаурус для автоматического концептуального индексирования как особый вид тезауруса

Отличительные особенности тезауруса для автоматического концептуального индексирования.

Основной целью разработки традиционных информационно-поисковых тезаурусов является использование их единиц (дескрипторов) для описания основных тем документов в процессе ручного индексирования. По своей сути тезаурус для ручного индексирования является искусственным языком описания, построенным на основе естественного языка. При этом сам процесс индексирования по такому тезаурусу базируется на лингвистических, грамматических знаниях, а также знаниях о предметной области, которые имеются у профессиональных индексаторов текстов. Индексатор сначала должен прочитать текст, понять его и затем изложить содержание текста, пользуясь дескрипторами, указанными в информационно-поисковом тезаурусе. Индексатор должен хорошо понимать всю терминологию, использованную в тексте, - для описания основной темы текста ему понадобится значительно меньшее количество терминов.

При автоматической обработке текстов человека-посредника между текстом и описанием его содержания в виде дескрипторов нет. Есть только автоматический процесс и Тезаурус, который должен содержать и те знания, которые содержатся в традиционных информационно-поисковых тезаурусах, и те знания (насколько это возможно), которые использует индексатор для определения основной темы текста.

Именно поэтому традиционные тезаурусы, разработанные для ручного индексирования, невозможно использовать при автоматическом индексировании.

Разработка тезауруса для автоматического индексирования (далее – АИ-тезауруса) характеризуется, прежде всего, необходимостью описания значительно большего количества слов и словосочетаний, встречающихся в текстах данной предметной области. АИ-тезаурус должен включать не только термины, которые представляют важные понятия в текстах данной предметной области, но также охватывать широкий круг более специфических терминов, обнаружение которых в конкретном тексте сделает

этот текст релевантным запросу по понятиям более высокого уровня, например, должны быть описаны не только дескриптор *РЫБА* и его основные подразделения, такие как *МОРСКИЕ РЫБЫ*, *АНАДРОМНЫЕ РЫБЫ* и т.п., но и значительное количество конкретных видов рыб с тем, чтобы текст, обсуждающий проблемы вылова минтая, мог бы быть получен при поиске по термину *рыба*.

Синонимические ряды понятий должны быть значительно богаче, чем совокупности вариантов дескриптора в тезаурусе для ручного индексирования, поскольку синонимы должны описывать различные способы выражения данного понятия в тексте для автоматического процесса, а не для человека. Ряды синонимов включают в себя не только существительные и именные группы, а также прилагательные, глаголы, глагольные группы. Расширение терминологической базы АИ-тезауруса ведет к необходимости описания многозначных терминов.

Расширение понятийной базы тезауруса ведет к увеличению и усложнению функций отношений между понятиями тезауруса (концептуальными отношениями): возникает необходимость логического вывода отношений, поскольку описать отношения всех дескрипторов со всеми близкими дескрипторами АИ-тезауруса становится трудоемким занятием и затрудняет проверку таких описаний.

Общественно-политический тезаурус как ресурс для автоматического концептуального индексирования текстов

С 1994 года началась разработка Общественно-политического информационно-поискового тезауруса как ресурса для автоматического индексирования. Общественно-политический тезаурус включает в себя терминологию экономической, политической, военной, финансовой, законодательной, социальной, культурной и других сфер деятельности - терминологию, которая встречается в таких общезначимых документах как официальные и законодательные документы, международные договоры, сообщения информационных агентств и газетные публикации.

Общественно-политический тезаурус представляет собой иерархическую сеть понятий, каждое из которых имеет ряд текстовых вариантов (способов языкового выражения) и совокупность отношений с другими понятиями тезауруса.

Предметная область тезауруса - это широкая политематическая область современных общественных отношений, проблем современного общества. Поэтому набор понятий Тезауруса соответствует понятийному содержанию и нормативных документов, и газетным публикациям, и в значительной степени научным публикациям по общественным наукам.

С 1995 года Общественно-политический тезаурус активно и успешно применяется для различных приложений автоматической обработки текстов, таких как автоматическое концептуальное индексирование, автоматическое рубрицирование с использованием нескольких рубрикаторов, автоматическое аннотирование текстов.

В настоящее время Общественно-политический тезаурус включает 33 тысячи понятий, 87 тысяч русскоязычных слов, терминов, выражений, 130 тысяч отношений между понятиями.

Отношения в информационно-поисковых ресурсах: альтернативы

Современные подходы к описанию отношений при разработке онтологий

Рассмотрим онтологию, предназначенную для работы в информационно-поисковых задачах, и содержащую описания понятий предметной области, отношения между понятиями задаются в виде предикатов. Пусть свойства отношений (аксиомы вывода) описываются как правила вида: *if* $P(x_1,...x_n)$ *then* $Q(y_1... y_m)$.

Чтобы инициализировать эти правила, необходимо быть уверенным, что $P(x_1,...x_n)$ определяется с высокой точностью. При современном уровне систем автоматической обработки текстов в большой разнородной коллекции не для любых типов предикатов $P(x_1,...x_n)$ можно гарантировать приемлемый уровень точности и полноты их нахождения.

Например, различные аргументы предиката $P(x_1,...x_n)$ могут оказаться в разных частях длинного предложения, что значительно усложнит сборку предиката, или в разных предложениях текста, например, из-за использования эллиптической конструкции или местоимения и т.п. Проблемы с правильной идентификацией аргументов предикатов в текстах могут свести к нулю возможные преимущества применения знаний, описанных в онтологиях, по сравнению с пословным поиском.

Среди потенциального множества отношений понятия наиболее стабильно можно опираться на те отношения, которые не исчезают, не меняются в течение всего срока существования любого или подавляющего большинства экземпляров понятия. Например, любой лес всегда состоит из деревьев.

Наиболее известным типом отношения, которое выполняется для всех экземпляров, является таксономическое отношение. Так, если *C1* упомянуто в тексте и *C1* является видом *C2*, это означает, что в тексте упомянуто и *C2*. Если данный текст релевантен запросу о *C1*, то он будет релевантен и запросу о *C2*.

В условиях невозможности использования сложных правил вывода, для осуществления вывода по тексту желательно найти другие типы отношений, обладающие свойствами транзитивности и наследования, подобно таксономическим отношениям. Проблема рассмотрения взаимного сосуществования понятия является центральной в теории зависимости философской дисциплины «формальная онтология».

Отношения онтологической зависимости

Как известно, в течении столетий в рамках философии существовала отдельная отрасль знаний, называемая Онтология – учение о бытии. В рамках этого направления развивается Формальная Онтология – учение, изучающее сущности с точки зрения их формальных свойств (. Одним из основных инструментов анализа сущностей в рамках Формальной Онтологии является теория зависимости (Guarino, 1998, Gangemi et.al. 2001).

Главным вопросом теории зависимости является такой: может ли сущность (*C1*) существовать сама по себе, или подразумевает существование чего-либо еще (*C2*):

- подразумевает ли существование сущности существование чего-либо какой-либо конкретной сущности (строгая зависимость - rigid dependence), например, *кипение (C1)* невозможно без существования конкретного объема жидкости (*C2*), которая кипит;
- предполагается ли существование примеров некоторого класса (generic dependence – зависимость от класса) некоторых сущностей, например, возникновение понятия *гараж (C1)* невозможно без существования понятия *автомобиль (C2),* хотя конкретный гараж может строится безотносительно к конкретному автомобилю.
- предполагает ли существование *C1* в некоторый момент времени t_1, существования *C2* в некоторый другой момент времени t_2 (историческая зависимость), например, понятие *солома (C1)* исторически зависит от понятия *молотьба (C2)*, поскольку солома не может возникнуть без предварительного процесса молотьбы, вместе с тем эти работы заканчиваются, а солома длительное время продолжает существовать.

Перечисленные выше типы отношений онтологической зависимости упорядочены по мере снижения объема пересечения сфер существования зависящего понятия и главного понятия.

При строгой зависимости зависимое понятие не может быть оторвано от конкретного экземпляра главного понятия, поэтому если возникает,

существует, обсуждается конкретный пример такого жестко зависимого понятия, то существует и обсуждается пример главного понятия.

В случае зависимости по классу конкретный пример зависимого понятия может быть оторван от главного понятия, с ним может происходить что-то, не связанное с главным понятием, но обычно недолго и в относительно небольшой доле примеров зависимого понятия, например, в гараже может быть совершено преступление, и оно может не иметь никакого отношения к автомобилям.

При исторической зависимости пример зависимого понятия может достаточно долго существовать без главного понятия и участвовать в самых разных ситуациях, например, *сельскохозяйственная продукция* создается в процессе *сельскохозяйственного производства*, затем продукция значимое время живет «своей жизнью»: перевозится, продается, хранится.

Подход к описанию отношений в Общественно-политическом тезаурусе

Набор отношений специально подобран для эффективной работы в информационно-поисковых приложениях. Имеется четыре основных типа отношения

Первый тип отношений – родовидовое отношение НИЖЕ-ВЫШЕ, обладает свойством транзитивности и наследования.

Второе тип отношений – отношение ЧАСТЬ-ЦЕЛОЕ. Используется не только для описания физических частей, но и для других внутренних сущностей понятия, таких как свойства или роли для ситуаций. Важным условием при установлении этого отношения является то, что понятия-части должны быть жестко связаны со своим целым, то есть каждый пример понятия-части должен в течение всего времени своего существования являться частью для понятия-целого, и не относиться к чему-либо другому.

В этих условиях удается выполнить свойство транзитивности введенного таким образом отношения ЧАСТЬ-ЦЕЛОЕ, что очень важно для автоматического вывода в процессе автоматической обработки текстов.

Еще один тип отношения, называемого несимметричной ассоциацией АСЦ2-АСЦ1, связывает два понятия, которые не могут быть связаны выше рассмотренными отношениями, но когда одно из которых не существовало бы без существования другого. Например, понятие *САММИТ* требует существования понятия *ГЛАВА ГОСУДАРСТВА*. Последний тип отношений – симметричная ассоциация связывает, например, понятия очень близкие по смыслу, но которые разработчики не решились склеить в одно понятие.

Отношения ВЫШЕ-НИЖЕ, ЧАСТЬ-ЦЕЛОЕ и несимметричная ассоциация являются иерархическими отношениями. Таким образом, на основе свойств иерархичности, транзитивности и наследования для каждого

понятия может быть определена совокупность понятий, которые являются для него нижестоящими понятиями по иерархии.

Таким образом, два отношения в тезаурусе из четырех существенно связаны с понятием онтологической зависимости. В количественном отношении эти два отношения занимают приблизительно половину из всех отношений тезауруса.

Нарушение условий надежности

Если условия надежности выполняются почти всегда, по умолчанию, то используются специальные пометки – модификаторы отношений, что означает, что отношение более слабое. В связи с этим вводятся ограничения по транзитивности.

Используются два модификатора:

- Модификатор B («возможно») - отношение выполняется не для всех примеров
- Модификатор A («аспект», точка зрения) – отношение существует не все время

ПЕНСИОНЕР	
ВЫШЕ_B	*СТАРЫЙ ЧЕЛОВЕК*
ЦЕЛОЕ_A	*ПЕНСИОННАЯ СИСТЕМА*

Сравнение используемого отношения ЧАСТЬ-ЦЕЛОЕ с другими подходами

Наиболее близким по такой трактовке отношений ЧАСТЬ-ЦЕЛОЕ является онтология Sowa. Рассматривая классификацию ролей и отношений в своей онтологии верхнего уровня, автор рассматривает случаи зависимости понятий друг от друга prehension. Зависимость может быть внешняя (extrinsic) и внутренняя (intrinsic). Если одна сущность в отношении зависимости может исчезнуть, не меняя форму или существование другой сущности, это означает, что это отношение внешнее. Если исчезновение одной из сущности меняет структуру или существование другой сущности, то это отношение внутреннее.

Основным видом внутренне зависимых сущностей являются компоненты (Intrinsic Prehended Entity). Среди компонентов выделяются части и свойства. Части делятся на физические части, участников и стадии, а свойства делятся на атрибуты и способы.

Таким образом, Sowa также объединяет в один куст отношений такие отношения как физические части, участники, стадии, свойства по свойству внутренней зависимости.

Разработка Общественно-политического тезауруса как ресурса для автоматической обработки текстов как соединение трех существующих традиций. Общественно-политический тезаурусу как лингвистическая онтология

Таким образом Общественно-политический тезаурус представляет собой лингвистическую онтологию, которая строится на сочетании трех различных традиций и методологий:

- методологии разработки традиционных информационно-поисковых тезаурусов;
- методологии разработки лингвистических ресурсов типа WordNet (Принстонский университет);
- методологии созданий формальных онтологий.

Поскольку предполагается работать с терминологией, большими предметными областями и свободными текстами, то важно использовать опыт разработки информационно-поисковых тезаурусов, а именно:

- информационно-поисковый контекст;
- единицы онтологии создаются на основе значений терминов;
- описание большого числа многословных выражений, принципы включения (невключения) многословных единиц;
- небольшой набор отношений между понятийными единицами.

Так как предполагается использовать онтологию в автоматическом режиме обработки текстов, то необходимо использовать методологию разработки лексических ресурсов типа WordNet, в которой важны следующие положения:

- понятия онтологии создаются на основе значений реально существующих языковых выражений, терминов;
- многоступенчатое иерархическое построение лексико-терминологической системы понятий;
- принципы описания значений многозначных слов и выражений.

Из методологии разработки формальных онтологий важны следующие положения:

- разработка лингвистической онтологии как иерархической системы понятий;
- использование для описания нетаксономических отношений понятий отношений онтологической зависимости, которые описывают зависимость существования понятия или примеров понятия от существования других понятий (примеров понятия). В [Лукашевич, 2004] показано, что применение

таких отношений в лингвистическом ресурсе эффективно для решения задач информационного поиска;

в качестве аксиом (правил вывода) использование свойств транзитивности и наследования таксономических отношений и транзитивности отношений онтологической зависимости.

Вопросы к лекции

1. В чем состоят отличительные особенности Тезауруса для автоматического концептуального индексирования?
2. Каковы возможные способы установление отношений в тезаурусах?
3. Что такое отношения онтологической зависимости?

Литература

1. Лукашевич Н.В., Салий А.Д., Тезаурус для автоматического рубрицирования и индексирования: разработка, структура, ведение // НТИ. Сер.2. - 1996. - N 1. - С.1-6.
2. Лукашевич Н.В., Добров Б.В., Тезаурус для автоматического концептуального индексирования как особый вид лингвистического ресурса // Труды международного семинара Диалог-2001. - Аксаково-2001.- с.273-279.
3. Лукашевич Н.В, Добров Б.В., Отношения в онтологиях для решения задач информационного поиска в больших разнородных текстовых коллекциях // Девятая национальная конференция по искусственному интеллекту с международным участием КИИ-2004 (28 сентября –2 октября 2004 г., Тверь) : Труды конференции. В 3-х т. - Т2. – М.: Физматлит, 2004. – С.544-551.
4. Gangemi, Aldo, Nicola Guarino, and Alessandro Oltramari, 2001. Conceptual analysis of lexical taxonomies:the case of wordnet top-level. In *Proceedings of the international conference on Formal Ontology in Information Systems*. ACM Press.
5. Guarino N., Some Ontological Principles for Designing Upper Level Lexical Resources. // Proceedings of First International Conference on Language Resources and Evaluation, 1998.
6. Madsen B, Pedersen B., Thomsen H., (2000). Semantic Relations in Content-Based Querying Systems – a Research Presentation from the OntoQuery Project // Proceedings of 1st International Workshop, OntoLex 2000. – p. 72-81.
7. Soergel D., Lauser B., Liang A., Fisseha F., Keizer J., Katz S. Reengineering Thesauri for New Applications: the AGROVOC Example. - Article No. 257, 2004-03-17.

8. Sowa J., *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA, ©2000.
9. Tudhope D., Alani H., Jones Cr. Augmenting Thesaurus Relationships: Possibilities for Retrieval. – Journal of Digital Libraries. Volume 1, Issue 8. – 2001

10.2. Тезаурус для автоматического концептуального индексирования как ресурс для решения информационно-поисковых задач

С 1995 года Общественно-политический тезаурус используется в таких областях автоматической обработки текстов как автоматическое концептуальное индексирование, автоматическая рубрикация текстов, автоматическое аннотирование текстов. Все эти применения тезауруса базируются на разработанном авторами тематическом представлении текста.

Тезаурус и технология автоматического построения тематического представления содержания документа позволили развить в рамках УИС РОССИЯ гибкую технологию эффективной автоматической рубрикации текстов. Наши системы автоматической рубрикации работают с такими рубрикаторами как рубрикатор исследовательской службы конгресса США, общеправовым тематическим классификатором Центральной избирательной комиссии РФ, классификатором правовых актов РФ. Всего было внедрено шесть различных систем автоматической рубрикации с разными рубрикаторами размером от 35 до 1200 рубрик.

Знания, описанные в Тезаурусе, а также технология построения тематического представления позволили создать систему автоматического аннотирования текстов, основанную на знаниях. В 1998 году с нашей программой автоматического аннотирования англоязычных текстов мы участвовали в соревнованиях в рамках конференции SUMMAC, где эта программа получила лучшие результаты в номинации <Индикативная аннотация наилучшей длины>.

Тезаурус используется как инструмент для автоматического концептуального индексирования и ранжированного информационного поиска в Университетской информационной системе РОССИЯ (www.cir.ru).

АЛОТ: основные этапы

На первом этапе работы алгоритма происходит сравнение единиц текста с единицами Тезауруса. Сравнение текста и Тезауруса происходит на основе морфологического представления единиц текста и единиц Тезауруса. Из

множества найденных в тексте единиц Тезауруса выбирается единица, имеющая максимальную длину. Если один и тот же фрагмент текста соответствует разным единицам Тезауруса, то фиксируется многозначность термина.

В результате сопоставления с Тезаурусом текст отражается в последовательность дескрипторов Тезауруса. Все синонимы (варианты) одного и того же дескриптора отображаются в соответствующий дескриптор и далее не различаются. Для каждого дескриптора фиксируется частота его встречаемости в тексте.

Чтобы определить тезаурусные связи между дескрипторами текста, необходимо найти те пары дескрипторов, которые описаны в тезаурусных статьях друг друга. Но этого недостаточно. Необходимо также найти и такие пары дескрипторов текста, связи между которыми выводятся по свойствам транзитивности и наследования. Совокупность связанных между собой дескрипторов текста, полученных в результате применения процедуры вывода, называется проекцией Тезауруса на текст (тезаурусной проекцией).

Таким образом, два дескриптора текста оказываются непосредственно связанными в тезаурусной проекции, если:

1) дескрипторы текста находятся в одной тезаурусной статье;
2) между дескрипторами текста существует путь, состоящий только из последовательности связей ВЫШЕ, ЦЕЛОЕ (по транзитивности связей ВЫШЕ и ЦЕЛОЕ);
3) между дескрипторами существует путь, состоящий из последовательности связей ВЫШЕ, ЦЕЛОЕ и одной связи АССОЦИАЦИЯ (по свойству наследования связи АССОЦИАЦИЯ связями ВЫШЕ и ЦЕЛОЕ).

В построении тезаурусной проекции равным образом участвуют все дескрипторы, соответствующие неоднозначному термину. На основе тезаурусной проекции производится выбор дескриптора, соответствующего определенному значению термина. Для каждого значения неоднозначного термина проверяется :

- употреблялись ли в данном тексте наряду с неоднозначным термином однозначные термины, соответствующие дескриптору, выражающему это значение неоднозначного термина;
- имеет ли дескриптор, соответствующий этому значению неоднозначного термина, тезаурусные связи с другими дескрипторами проекции.

Если выполняется одно из вышеперечисленных условий, то считается, что "текст поддерживает" данное значение неоднозначного термина. Если текст "поддерживает" только одно значение неоднозначного термина, то выбирается соответствующий ему дескриптор.

Если текст "поддерживает" дескрипторы, соответствующие разным значениям термина, то для каждого вхождения неоднозначного термина рассматриваются ближайшие по тексту дескрипторы, для них проверяются вышеуказанные условия и выбирается тот дескриптор неоднозначного термина, который "поддерживается" первым из ближайших по тексту

дескрипторов.

Теоретические основы построения тематического представления

Популярной теорией в области автоматической обработки текстов является теория Ван Дейка, Кинча, которые указывают, что основная тема текста может быть описана некоторой пропозицией. В работе (Ван Дейк, Кинч, 1988) такая пропозиция называется макропропозицией. Основное содержание текста может быть представлено как иерархическая структура в том смысле, что тема всего текста может быть обычно описана посредством более конкретных тем текста, которые в свою очередь могут быть охарактеризованы посредством еще более конкретных подтем и т.п. Каждое предложение связного текста посвящено раскрытию той или иной подтемы основной темы текста.

Второе направление исследований базируется на автоматическом выявлении лексической связности текста. Какими бы отношениями не были связаны между собой предложения текста, часть слов, входящих в состав этих предложений, должны быть лексически связаны между собой, причем эти отношения между словами заранее известны автору и читателю текста.

Действительно, формулировка основной темы текста содержит некоторую совокупность слов, наиболее значимых для передачи содержания текста. Если рассмотреть текст, то можно видеть, что слова, близкие по смыслу к словам основной темы, образуют лексические цепочки, которые пронизывают весь текст. Естественно предположить, что если имеется лингвистический ресурс, в котором описаны разнообразные смысловые связи между словами, то можно двигаться по тексту, находить связанные по смыслу слова, формировать лексические цепочки (Morris J., Hirst G. 1991). Самые частотные (или выделенные по другим критериям) цепочки могли бы показать, чему именно посвящен конкретный текст (Barzilay R., Elhadad M. 1997).

Предположим, что мы сформулировали основную тему некоторого текста. В ней упомянуты некоторые понятия и/или конкретные объекты

текста. Подтемы текста раскрывают взаимоотношения между этими основными понятиями/объектами и поэтому должны тем или иным образом ссылаться на них, используя повторы слов, синонимы или другие слова, семантически связанные с понятиями основной темы (далее *основные понятия* текста). Таким образом, основным понятиям текста соответствуют некоторые совокупности слов текста (и совокупность понятий, стоящими за этими словами), которые используются в данном тексте для ссылки на эти основные понятия. Такие совокупности слов обычно пронизывают весь текст "красной нитью"

Если подтема текста раскрывается в более специфических подтемах, то для ссылки на основные понятия этой подтемы в свою очередь возникают более короткие "нити" слов. Таким образом, начало лексической цепочки нужно связывать не с началом текста, а с наиболее важным для содержания текста понятием, который должен стать центром этой цепочки, а все элементы лексической цепочки должны быть, прежде всего, связаны лексическим отношением именно с этим центром (последователями подхода на основе WordNet также обсуждается необходимость нахождения центра лексической цепочки, правда, уже после ее формирования).

Мы предполагаем, что более важные для текста понятия чаще всего так или иначе выделены в тексте относительно других близких им по смыслу понятий (например, частотностью, упоминанием в заголовке текста). Структуру, в которой все понятия связаны по тезаурусу с одним и тем же понятием, мы называем тематическим узлом, а главное понятие тематического узла – тематическим центром. Собственно расположение в тексте слов, соответствующих этим тематическим узлам, и создает эффект лексических цепочек.

Построение тематических узлов

Каждая тема, обсуждаемая в тексте, выражается обычно не одним термином, а совокупностью тематически близких терминов. Например, тема науки может развиваться в тексте посредством следующих терминов: *математика, физика, прикладное исследование, фундаментальное исследование, научный работник*. Тот термин, который наиболее точно характеризует развиваемую в тексте тему, обычно некоторым образом выделен из всей совокупности тематически близких терминов: такой термин может быть употреблен в заголовке и/или в начале текста, иметь максимальную частотность среди других тематически близких терминов.

Главным термином темы может стать любой термин Тезауруса, независимо от его уровня общности/специфичности. Так, главным термином темы текста может стать термин математика, если речь в тексте идет о

развитии математики; или главным термином может стать термин научный работник, если речь в тексте идет об оплате труда научных работников или о выезде ученых за рубеж.

Совокупность тематически связанных между собой дескрипторов с выделенным среди них главным дескриптором называется тематическим узлом.

Тематическая связанность терминов отображается в связях между соответствующими дескрипторами в тезаурусной проекции. Тезаурусная проекция обычно состоит из нескольких фрагментов связанных между собой дескрипторов. Каждый такой связный фрагмент может иметь достаточно сложную структуру, и в него могут входить далекие друг от друга дескрипторы. Таким образом, чтобы выделить тематические узлы необходимо провести дополнительное разбиение тезаурусной проекции.

Как показали наши эксперименты, наиболее эффективно проводить разбиение следующим образом:

Создание тематического узла начинается с выбора главного дескриптора тематического узла. Сначала тематические узлы собираются вокруг дескрипторов заголовка и первого предложения текста. Затем тематические узлы собираются для остальных дескрипторов, начиная с самых частотных. Те дескрипторы, которые уже попали в тематический узел некоторого дескриптора, свой тематический узел не образуют.

После того, как выбран главный дескриптор очередного тематического узла, в тематический узел включаются дескрипторы, непосредственно связанные с главным дескриптором тематического узла в тезаурусной проекции и дескрипторы, связанные с главным дескриптором посредством такой совокупности тезаурусных связей, которые можно свести к одной связи путем применения свойств транзитивности и наследования.

Выделение основных тематических узлов

Для выявления основных тематических узлов мы производим следующую процедуру:

- в процессе сопоставления текста с тезаурусом для каждого понятия тезауруса, найденного в тексте, запоминаются его соседи-понятия влево и вправо. В экспериментах было получено, что фиксация трех соседей вправо и влево представляется оптимальной. Такие пары понятий текста мы называем текстовыми связями понятия. Знак абзаца прерывает набор текстовых связей;
- текстовые связи разных вхождений понятия в тексте суммируются. В результате мы получаем частотность текстовых связей понятий между собой;

- в процессе создания тематических узлов текстовые связи каждого понятия в узле суммируются и получаются текстовые связи тематических узлов;
- выбираются три тематических узла, суммарная частотность попарных текстовых связей между которыми, является максимальной среди других треугольников текста. Это и есть первая тройка основных тематических узлов в тексте, центры этих основных тематических узлов являются элементами основной темы документа;
- далее необходимо проверить, нет ли еще элементов основной темы документа. Для этого среди оставшихся тематических узлов ищутся тематические узлы, которые имеют текстовые связи как с уже полученными основными тематическими узлами, так и между собой;
- таким образом, совокупность основных тематических узлов получена. Их текстовые связи образуют между собой симплексы – фигуры, в которых каждая вершина имеет ребро с другой вершиной (треугольник, пирамида и т.п.). Важная особенность выявления основных тематических узлов на основе симплексов текстовых связей – независимость процесса от размера, жанра и языка обрабатываемых текстов.

Эксперименты показали, что тематическое представление может быть построено для текстов любого размера и разнообразных типов. Тематические представления были построены для более 100 Мб официальных документов Российской Федерации, международных договоров, для большинства российских законов 1990-1997 гг. Тематические представления были также построены для более 50 Мб сообщений информационных агентств и газетных статей. Размеры обрабатываемых документов варьировались от 500 байт до более 500 Кб. (Гражданский Кодекс Российской Федерации, Таможенный кодекс Российской Федерации).

Тезаурус как поисковых механизм УИС Россия:

Тезаурус существенно используется в интерфейсе УИС РОССИЯ для следующих задач терминологического поиска:

- уточнения запроса, когда выбор более точного термина позволяет получать только требуемые документы, например, выбирая вместо всех типов *СТРОИТЕЛЬСТВА* именно *ДОРОЖНОЕ СТРОИТЕЛЬСТВО* (*автодорожное строительство, дорожно-строительные работы, строительство дорог, строительно-дорожный* и т.д.);

- автоматического расширения запроса по синонимам (*НАЛОГОВАЯ СИСТЕМА == налоговый режим*), а также по иерархии (*МИГРАЦИЯ ---- БЕЖЕНЦЫ*, *ВЫНУЖДЕННЫЕ ПЕРЕСЕЛЕНЦЫ* и т.д.).

Структурная тематическая аннотация

Структурная тематическая аннотация представляет содержание текста посредством описания его основных тем, которые моделируются совокупностью терминов, относящихся к этим темам. Структурная тематическая аннотация содержит наиболее информативные фрагменты тематического представления текста, которое включает все термины текста, разбитые на тематические узлы и отношения между различными темами и подтемами текста.

Автоматическое построение связной аннотации

Знания человека о тематической связности между терминами вытекают из знаний о предметной области, в рамках которой написан текст. Таким образом, то новое и важное, что несет в себе текст и что должна отразить в себе аннотация, это именно то, каким образом взаимодействуют между собой разные основные темы текста. Отсюда следует первый принцип составления аннотаций: важными (информативными) и, следовательно, возможно включенными в аннотацию считаются те предложения текста, которые содержат по крайней мере два термина, входящих в состав разных основных тем текста.

Предложений, содержащих термины одних и тех же двух основных тем, в тексте может оказаться достаточно много. Для аннотации необходимо выделить одно предложение, в котором взаимодействие этих двух тем характеризуется “наилучшим образом”.

Чтобы понять, что значит “наилучшим образом” рассмотрим, как та или иная тема развивается в тексте. Не все основные темы начинают обсуждаться в тексте сразу, с первого предложения -- часть из них возникает в последующих предложениях. Чтобы сохранить связность и последовательность изложения текста автор именно в этом первом предложении новой темы должен наиболее точно указать связь новой темы со всем предшествующим текстом. Таким образом, второй принцип составлений аннотаций -- это для каждой пары выявленных основных тем текста выбрать в аннотацию те предложения, в которых эта пара обсуждалась первый раз, следуя по порядку текста.

Нужно отметить, что при хорошем покрытии предметной области Тезаурусом появление в очередном предложении новой темы выявляется

весьма точно, а это означает, что связность получаемой аннотации в среднем весьма высока.

Вопросы к лекции

1. Перечислите этапы автоматической обработки текстов на основе Тезауруса.
2. Как моделируется связность текста?
3. Каков принцип построения связной аннотации текста?

Литература

1. ван Дейк Т.А., Кинч В. 1988. Стратегии понимания связного текста. // Новое в зарубежной лингвистике. Вып. 23. - М.: Прогресс. - С.153-211.
2. Лукашевич Н.В., Добров Б.В., Тезаурус для автоматического концептуального индексирования как особый вид лингвистического ресурса // Труды международного семинара Диалог-2001. - Аксаково-2001.- с.273-279.
3. Лукашевич Н.В., Автоматическое построение аннотаций на основе тематического представления текста // Труды международного семинара Диалог'97. - Москва, 1997 - С. 188-191
4. Лукашевич Н.В., Добров Б.В., Построение структурной тематической аннотации текста // Труды международного семинара Диалог-98 - Том 2 – 1998 - С.795-802.
5. Barzilay R., Elhadad M. Using Lexical Chains for Text Summarization. - ACL/ EACL Workshop Intelligent Scalable Text Summarization.- Madrid, 1997.
6. Hirst G., St-Onge D., Lexical Chains as representation of context for the detection and correction malapropisms. // In C. Fellbaum, editor, WordNet: An electronic lexical database and some of its applications. Cambrige, MA: The MIT Press, 1997.

10.3. Технология автоматической рубрикации текстов с использованием тезауруса для автоматического концептуального индексирования

Задачей систем автоматического рубрицирования является разбиение поступающего потока текстов на тематические подпотоки в соответствии заранее заданными рубриками.

Дадим некоторые определения.

Под **рубрикатором** понимается классификационная таблица иерархической классификации, содержащая полный перечень включенных в систему классов и предназначенная для систематизации информационных фондов, массивов и изданий, а также для поиска в них (ГОСТ 7.74-96).

Предметная рубрика - элемент информационно-поискового языка, представляющий собой краткую формулировку темы на естественном языке.

Адекватная предметная рубрика - предметная рубрика, формулировка которой выражает объем понятия, наиболее точно соответствующего объему понятия о предмете документа.

При составлении рубрикатора:

каждый документ предметной области должен иметь соответствующую предметную рубрику;

не должно быть рубрик, которым соответствует относительно малое количество документов;

рубрики по возможности должны быть четко отделены друг от друга. Для близких по содержанию рубрик лучше иметь краткие комментарии, в каких случаях проставлять одну из рубрик, в каких случаях обе рубрики.

Критерии оценки качества рубрицирования

Для оценки эффективности работы систем рубрицирования используются такие характеристики, как полнота и точность.

Полнота - это отношение R/Q, где R - количество текстов, правильно отнесенных к некоторой рубрике, а Q - общее количество текстов, которые должны быть отнесены к этой рубрике.

Точность – это отношение R/L, где R - количество текстов, правильно отнесенных системой к некоторой рубрике, а L - общее количество текстов, отнесенных системой к этой рубрике.

Проблемы ручного рубрицирования

Характерными особенностями ручного рубрицирования являются:

- высокая точность рубрицирования. Обычно процент документов, в которых проставлена явно неправильная рубрика, чрезвычайно мал,
- низкая полнота рубрицирования. Обычно специалисты по рубрикации проставляют одну-две основных рубрики, характеризующие основное содержание документа, хотя документ может быть отнесен и к ряду других рубрик. В результате получается, что при сравнении результатов рубрикации разными экспертами одних и тех же документов процент совпадения проставленных рубрик может оказаться весьма низким – 60 %. В результате похожие документы

могут получить достаточно разные наборы рубрик. Такая ситуация усугубляется при увеличении величины и иерархической сложности рубрикатора. Непоследовательность ручного рубрицирования становится серьезной проблемой для настройки разного типа систем автоматического рубрицирования, поскольку затрудняется построение формальных правил отнесения документов к той или иной рубрике.

- низкая скорость обработки документов.

Методы автоматической рубрикации

Наиболее эффективными, но и наиболее трудозатратными, является методы автоматического рубрицирования, основанные на знаниях. При рубрицировании текстов на основе знаний используются заранее сформированные базы знаний, в которых описываются языковые выражения, соответствующие той или иной рубрике, правила выбора между рубриками и др.

Другим классом методов для автоматической рубрикации текстов являются методы машинного обучения, которые в качестве обучающих примеров используют заранее отрубрицированные вручную тексты.

Приводятся очень высокие оценки результатов работы методов машинного обучения, время обучения составляет доли секунд. Однако при ближайшем рассмотрении оказывается, что практически все такие методы тестируются на одной и той же текстовой коллекции -- это коллекция финансовых сообщений информационного агентства Рейтер [1], которая была специально создана несколько лет назад для тестирования методов автоматической рубрикации текстов.

Эта коллекция характеризуется следующими основными чертами:

1) рубрикатор, включающий 135 рубрик, относительно прост, без иерархии;
2) небольшие по величине тексты принадлежат достаточно узкой области финансовых известий;
3) для обучения представляется более 15 тысяч отрубрицированных документов;
4) подавляющее большинство документов относится к приблизительно 20 рубрикам рубрикатора.

Все эти особенности коллекции значительно упрощают решение задачи машинного обучения автоматической рубрикации текстов.

Проблемы автоматического рубрицирования

Проблемы автоматического рубрицирования связаны со следующими обстоятельствами:

1. для автоматической рубрикации нужно сначала так или иначе создать образ рубрики, как некоторое выражение на основе слов и (или) терминов реальных текстов. Это может быть сделано на основе экспертного описания рубрики или методов машинного обучения по уже отрубрицированных коллекций
2. при автоматической обработке конкретных текстов могут возникнуть достаточно серьезные проблемы анализа языкового материала, контекста употребления того или иного слова, требующие привлечения обширных знаний о языке и предметной области, которые очень трудно описать в действующих программных система автоматической рубрикации.

Типы ошибок автоматического рубрицирования

1. Появление «Лишних рубрик»

Содержание рубрики сложнее, чем это выглядит по формулировке

1.2. Лексическая многозначность

Текст отнесен не к той рубрике из-за того, что некоторые слова, сопоставленные рубрике, в конкретной тексте употреблены в другом значении, таком значении, которое не соответствует данной рубрике.

1.3. Ложная корреляция

может возникнуть в случаях, когда для отнесения текста к рубрике необходимо присутствие в тексте двух логических элементов. Например, для рубрицирования по рубрике «Экономические реформы» необходимо присутствие в тексте двух тематических элементов – темы экономики и темы реформы. Ложная корреляция и, соответственно, неправильное отнесение текста к данной рубрике возникает в тех случаях, когда такие тематические элементы присутствуют в тексте, но не имеют отношения друг к другу, например, такая ситуация может произойти, если в тексте речь шла о судебной реформе и были упомянуты некоторые экономические вопросы.

1.4. Рубрикация по несущественному элементу

Текст отнесен к рубрике по слову или словосочетанию, которое по сути соответствует содержанию рубрики, но в данном тексте это опорное слово

или словосочетание употреблено случайно или, в каком-то специфическом контексте, из-за чего текст становится нерелевантным рубрике.

2. Пропуск правильных рубрик

2.1. Нехватка базы описания рубрики

Правильная рубрика не определена, поскольку в тексте упомянуты слова, не описанные в словаре системы рубрицирования.

2.2. Лексическая многозначность

может стать причиной потери правильной рубрики для рубрикации.

2.3. Слишком сложная структура документа

может привести к пропуску правильной рубрики при автоматической рубрикации, например, если в состав документа входит заголовок и большая таблица, при этом все информация для правильного отнесения текста к рубрике содержится только в заголовке.

Методы машинного обучения в задачах рубрикации

В задаче рубрикации текстов используются различные методы машинного обучения:

- Отсечение по центрам тяжести
- Отсечение по ближайшим соседям (kNN)
- Отсечение по ближайшим точкам (SVM)
- Оптимальный линейный сепаратор SVM (Support Vector Machines)

Сложные задачи автоматической рубрикации текстов

Реальные задачи рубрикации текстов в значительной мере отличаются от задачи классификации сообщений на тестовой коллекции агентства Рейтер. На практике, если перед достаточно большой компанией встает задача автоматической рубрикации текстов, то обычно используются автоматизированные технологии, основанные на ручном подборе лексики под каждую рубрику рубрикатора с последующим контролем результатов рубрицирования.

- Множество примеров отсутствует и не может быть создано в короткое время
- Множество примеров существует, но отсутствовали требования к качеству
- Множество примеров противоречиво и недостаточно для большинства рубрик (очень большие классификаторы)

- Множество примеров для обучения из другой коллекции

Применение тезауруса для решения сложных задач рубрикации

В информационной системе УИС РОССИЯ реализована система автоматического рубрицирования, способная рубрицировать тексты различных типов (официальные документы, сообщения информационных агентств, газетные статьи), ее легко можно настроить на новый рубрикатор и новые типы текстов, рубрицирование можно осуществлять сразу по нескольким рубрикаторам.

Реализация такой гибкой технологии автоматического рубрицирования основывается на определенных способах представления знаний о предметной области и представления текстовой информации:

- знания о предметной области хранятся в виде иерархической сети в так называемом Тезаурусе по общественно-политической жизни России.
- рубрикаторы связываются с Тезаурусом посредством небольшого числа опорных терминов, рубрики остальных терминов выводятся по связям внутри Тезауруса, что стало возможным благодаря тщательной предварительной разработке тезаурусных связей, максимально полному отражению различных аспектов описываемых понятий;
- возможность обрабатывать тексты разных типов и размеров базируется на тематическом представлении содержания текста, которое моделирует основную тему и подтемы документа наборами (тематическими узлами) близких по смыслу терминов документа.

Схема описания рубрики

Каждая рубрика R описывается дизъюнкцией альтернатив, каждый дизъюнкт представляет собой конъюнкцию:

$$R = \bigcup_i D_i ; \qquad D_i = \bigcap_j K_{ij} ,$$

Конъюнкты в свою очередь описываются экспертами с помощью так называемых «опорных» понятий тезауруса. Для каждого опорного понятия задается правило его расширения $f(\cdot)$, определяющее каким образом вместе с опорным понятием учитывать подчиненные ему по иерархии понятия. Выделяются три случая – без расширения (обозначается символом «N»), полное расширение по дереву иерархии тезауруса (символ «E») и расширении только по родо-видовым связям (символ «L»).

Опорный концепт может быть как «положительным», который добавляет нижерасположенные понятия в описание конъюнкта, так и «отрицательным»,

который вырезает свои подчиненные понятия. Последовательность учета положительных и отрицательных опорных понятий регулируется заданием специального атрибута. Результатом применения расширения опорных понятий является совокупность понятий тезауруса, полностью описывающая конъюнкт:

$$K_{ij} = \bigcup_m f_m(c_{ijm}) \setminus \bigcup_n f_n(e_{ijn}) = \bigcup_k d_{ijk} .$$

Вопросы к лекции

1. Перечислите методы автоматической рубрикации
2. По каким причинам возникают сложности в задачах автоматической рубрикации текстов?
3. Какие рубрикаторы Вам известны? Опишите их характеристики.

Литература

1. Агеев М.С., Добров Б.В., Лукашевич Н.В., Поддержка системы автоматического рубрицирования для сложных задач классификации текстов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. Труды шестой Всероссийской научной конференции. Пущино, 29.09-01.10.2004 – Ин-т мат. проблем биологии, Пущино. – 2004. - С.216-225
2. Агеев М.С., Добров Б.В., Лукашевич Н.В., Сидоров А.В., Экспериментальные алгоритмы поиска/классификации и сравнение с «basic line» // Российский семинар по оценке методов информационного поиска. Труды второго российского семинара РОМИП'2004 (Пущино, 01.10.2004) – СПб: НИИ Химии СПбГУ. – 2004. –
3. Российский семинар по оценке методов информационного поиска, РОМИП 2004, Пущино, 2004.
4. Шабанов В.И., Андреев А.М. Метод классификации текстовых документов, основанный на полнотекстовом поиске. РОМИП 2003.
5. Hayes Ph. Intelligent High-Volume Processing Using Shallow, Domain-Specific Techniques // Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval. New Jersey,1992. - P.227-242.
6. Joachims T., Text Categorization with Support Vector Machines: Learning with Many Relevant Features // Proceedings of ECML-98, 10th European Conference on Machine Learning, 1998.
7. Reuters-21578 text categorization test collection (www.daviddlewis.com/resources/testcollections/reuters21578)

8. Riloff E., Lehnert W. 1994. Information Extraction as a Basis for High Precision Text Classification. *ACM Transactions on Information Systems*, 12(3):296-333.
9. Wasson M., Classification Technology at LexisNexis // SIGIR 2001 Workshop on Operational Text Classification.

ПРИЛОЖЕНИЯ

Приложение 1. Иерархии классов и свойств онтологии в области культуры CIDOC CRM

Далее приводится полный список классов и свойств онтологии CRM.

Иерархия Классов

```
E1_CRM Сущность (CRM Entity)
- E53_Место (Place)
- E2_Временная Сущность (Temporal Entity)
- - E3_Состояние (Condition State)
- - E4_Период (Period)
- - - E5_Событие (Event)
- - - - E63_Начало Существования (Beginning of Existence)
- - - - - E81_Трансформация (Transformation)
- - - - - E65_Событие Создания (Creation Event)
- - - - - - E83_Создание Типа (Type Creation)
- - - - - E12_Событие Изготовления (Production Event)
- - - - - E67_Рождение (Birth)
- - - - - E66_Событие Постройки (Formation Event)
- - - - E64_Конец Существования (End of Existence)
- - - - - E68_Роспуск (Dissolution)
- - - - - E81_Трансформация (Transformation)
- - - - - E6_Разрушение (Destruction)
- - - - - E69_Смерть (Death)
- - - - E7_Деятельность (Activity)
- - - - - E9_Перемещение (Move)
- - - - - E8_Событие Приобретения (Acquisition Event)
- - - - - E13_Назначение Атрибута (Attribute Assignment)
- - - - - - E16_Событие Измерения (Measurement Event)
- - - - - - E14_Оценка Состояния (Condition Assessment)
- - - - - - E15_Назначение Идентификатора (Identifier
              Assignment)
- - - - - - E17_Назначение Типа (Type Assignment)
- - - - - E11_Событие Изменения (Modification Event)
- - - - - - E80_Удаление Части (Part Removal)
- - - - - - E79_Добавление Части (Part Addition)
- - - - - - E12_Событие Изготовления (Production Event)
- - - - - E65_Событие Создания (Creation Event)
- - - - - - E83_Создание Типа (Type Creation)
- - - - - E10_Передача Опеки (Transfer of Custody)
- - - - - E66_Событие Постройки (Formation Event)
- E52_Интервал Времени (Time-Span)
- E54_Размер (Dimension)
- E77_Постоянная Сущность (Persistent Item)
```

- - E51_Контакт (Contact Point)
- - - E45_Адрес (Address)
- - E41_Обозначение (Appellation)
- - - E44_Обозначение Места (Place Appellation)
- - - - E45_Адрес (Address)
- - - - E47_Пространственные Координаты (Spatial Coordinates)
- - - - E48_Название Места (Place Name)
- - - - E46_Определение Района (Section Definition)
- - - E75_Обозначение Концептуального Объекта (Conceptual Object Appellation)
- - - E35_Заголовок (Title)
- - - E49_Обозначение Времени (Time Appellation)
- - - - E50_Дата (Date)
- - - E82_Обозначение Актанта (Actor Appellation)
- - - E42_Идентификатор Объекта (Object Identifier)
- - E70_Вещь (Stuff)
- - - E72_Объект Права (Legal Object)
- - - - E18_Материальный Предмет (Physical Stuff)
- - - - - E19_Материальный Объект (Physical Object)
- - - - - - E22_Технической Объект (Man-Made Object)
- - - - - - - E84_Носитель Информации (Information Carrier)
- - - - - - E20_Биологический Объект (Biological Object)
- - - - - - - E21_Личность (Person)
- - - - - E26_Физический Признак (Physical Feature)
- - - - - - E27_Участок (Site)
- - - - - - E25_Искусственный Признак (Man-Made Feature)
- - - - - E24_Материальный Продукт Деятельности Человека (Physical Man-Made Stuff)
- - - - - - E78_Коллекция (Collection)
- - - - - - E22_Технической Объект (Man-Made Object)
- - - - - - - E84_Носитель Информации (Information Carrier)
- - - - - - E25_Искусственный Признак (Man-Made Feature)
- - - - E73_Информационный Объект (Information Object)
- - - - - E36_Визуальный Предмет (Visual Item)
- - - - - - E38_Изображение (Image)
- - - - - - E37_Пометка (Mark)
- - - - - - - E34_Надпись (Inscription)
- - - - - E29_Проект или Процедура (Design or Procedure)
- - - - - E33_Линвистический Объект (Linguistic Object)
- - - - - - E35_Заголовок (Title)
- - - - - - E34_Надпись (Inscription)
- - - - - E31_Документ (Document)
- - - - - - E32_Официальный Документ (Authority Document)
- - - E71_Продукт Деятельности Человека (Man-Made Stuff)
- - - - E24_Материальный Продукт Деятельности Человека (Physical Man-Made Stuff)

```
- - - - - E78_Коллекция (Collection)
- - - - - E22_Технической Объект (Man-Made Object)
- - - - - - E84_Носитель Информации (Information Carrier)
- - - - - E25_Искусственный Признак (Man-Made Feature)
- - - - E28_Концептуальный Объект (Conceptual Object)
- - - - - E55_Тип (Type)
- - - - - - E56_Язык (Language)
- - - - - - E57_Материал (Material)
- - - - - - E58_Единица Измерения (Measurement Unit)
- - - - - E73_Информационный Объект (Information Object)
- - - - - - E36_Визуальный Предмет (Visual Item)
- - - - - - - E38_Изображение (Image)
- - - - - - - E37_Пометка (Mark)
- - - - - - - - E34_Надпись (Inscription)
- - - - - E29_Проект или Процедура (Design or Procedure)
- - - - - E33_Линвистический Объект (Linguistic Object)
- - - - - - E35_Заголовок (Title)
- - - - - - E34_Надпись (Inscription)
- - - - - E31_Документ (Document)
- - - - - - E32_Официальный Документ (Authority Document)
- - - - E30_Право (Right)
- - E39_Актор (Actor)
- - - E21_Личность (Person)
- - - E74_Группа (Group)
- - - - E40_Юридическое Лицо (Legal Body)
E59_Простое Значение (Primitive Value)
- E60_Число (Number)
- E61_Строка (String)
- E62_Примитив Времени (Time Primitive)
```

Иерархия Свойств

```
P92F_запустило в эксплуатацию (brought into existence)
- P108F_произвело (has produced)
- P94F_создало (has created)
- - P135F_создало тип (created type)
- P98F_родило (brought into life)
- P95F_сформировало (has formed)
- P123F_повлекла появление (resulted in)
P94B_было создано (was created by)
- P135B_был создан (was created by)
P31F_изменило (has modified)
- P112F_уменьшило (diminished)
- P108F_произвело (has produced)
P11B_участвовал в (participated in)
- P14B_выполнял (performed)
- - P29B_получил опеку посредством (received custody
        through)
```

- - P28B_опека отдана посредством (surrendered custody through)
- - P22B_получил право собственности посредством (acquired title through)
- - P23B_право собственности отдано посредством (surrendered title through)
- P99B_была распущена (was dissolved by)
- P96B_дала рождение (gave birth)

P14F_выполнялась (carried out by)
- P23F_передало право собственности от (transferred title from)
- P29F_опеку получил (custody received by)
- P22F_передало право собственности (transferred title to)
- P28F_опеку отдал (custody surrendered by)

P92B_было пущено в эксплуатацию (was brought into existence by)
- P95B_была сформирована (was formed by)
- P108B_было произведено (was produced by)
- P98B_был рожден (was born)
- P123B_была результатом (resulted from)
- P94B_было создано (was created by)
- - P135B_был создан (was created by)

P31B_изменен (was modified by)
- P108B_было произведено (was produced by)
- P112B_было уменьшено (was diminished by)
- P110B_было увеличено (was augmented by)

P14B_выполнял (performed)
- P29B_получил опеку посредством (received custody through)
- P28B_опека отдана посредством (surrendered custody through)
- P22B_получил право собственности посредством (acquired title through)
- P23B_право собственности отдано посредством (surrendered title through)

P11F_имело участника (had participant)
- P14F_выполнялась (carried out by)
- - P23F_передало право собственности от (transferred title from)
- - P29F_опеку получил (custody received by)
- - P22F_передало право собственности (transferred title to)
- - P28F_опеку отдал (custody surrendered by)
- P96F_матерью (by mother)
- P99F_распустило (dissolved)

P93F_положило конец существованию (took out of existence)
- P13F_уничтожило (destroyed)
- P99F_распустило (dissolved)
- P124F_трансформировало (transformed)

- P100F_была смертью (was death of)
P93B_существование было прекращено (was taken out of existence by)
- P124B_было трансформировано (was transformed by)
- P100B_умер (died in)
- P99B_была распущена (was dissolved by)
- P13B_было уничтожено (was destroyed by)
P47F_идентифицируется (is identified by)
- P48F_имеет предпочтительный идентификатор (has preferred identifier)
P94F_создало (has created)
- P135F_создало тип (created type)
P47B_идентифицирует (identifies)
- P48B_является предпочтительным идентификатором (is preferred identifier of) P131F_идентифицируется (is identified by)
P99F_распустило (dissolved)
P71B_перечислено в (is listed in)
P27F_перемещен из (moved from)
P42F_назначило (assigned)
P16B_использовалась для (was used for)
P48F_имеет предпочтительный идентификатор (has preferred identifier)
P26F_перемещен в (moved to)
P40F_наблюдало размерность (observed dimension)
P13F_уничтожило (destroyed)
P100B_умер (died in)
P138F_представляет (represents)
P52F_имеет текущего владельца (has current owner)
P70B_документировано (is documented in)
P26B_было пунктом назначения для (was destination of)
P29B_получил опеку посредством (received custody through)
P87B_идентифицирует (identifies)
P55F_имеет настоящее местоположение (has current location)
P112B_было уменьшено (was diminished by)
P108F_произвело (has produced)
P52B_является текущим владельцем для (is current owner of)
P40B_наблюдалась в (was observed in)
P124F_трансформировало (transformed)
P80_конец ограничен (end is qualified by)
P38F_отменило назначение (deassigned)
P65B_показан при помощи (is shown by)
P35F_идентифицировало (has identified)
P50B_является текущим смотрителем для (is current keeper of)
P95B_была сформирована (was formed by)
P36F_зарегистрировал (registered)
P33F_использовало особую технику (used specific technique)

P95F_сформировало (has formed)
P135B_был создан (was created by)
P123B_была результатом (resulted from)
P17B_обусловило мотивировало (motivated)
P108B_было произведено (was produced by)
P48B_является предпочтительным идентификатором (is preferred identifier of)
P102F_имеет заголовок (has title)
P98B_был рожден (was born)
P41F_классифицирует (classified)
P42B_назначено посредством (was assigned by)
P23B_право собственности отдано посредством (surrendered title through)
P112F_уменьшило (diminished)
P38B_был отменен посредством (was deassigned by)
P87F_идентифицируется (is identified by)
P29F_опеку получил (custody received by)
P13B_было уничтожено (was destroyed by)
P100F_была смертью (was death of)
P110B_было увеличено (was augmented by)
P98F_родило (brought into life)
P16F_использовала вещь (used specific object)
P73F_имеет перевод (has translation)
P123F_повлекла появление (resulted in)
P50F_имеет текущего смотрителя (has current keeper)
P96F_матерью (by mother)
P129F_касается (is about)
P25F_перемещен (moved)
P71F_перечисляет (lists)
P28B_опека отдана посредством (surrendered custody through)
P78F_идентифицируется (is identified by)
P138B_имеет представление (has representation)
P102B_является заголовком (is title of)
P37B_было применено (was assigned by)
P34F_имела дело с (concerned)
P36B_был зарегистрирован (was registered by)
P136B_стало основой для создания типа (supported type creation)
P27B_было исходной точкой отправки для (was origin of)
P39F_измерил (measured)
P136F_был основан на (was based on)
P23F_передало право собственности от (transferred title from)
P55B_имеет на территории (currently holds)
P124B_было трансформировано (was transformed by)
P33B_была использована в (was used by)

```
P22B_получил право собственности посредством (acquired title
     through)
P22F_передало право собственности (transferred title to)
P39B_был измерен (was measured by)
P37F_назначило (assigned)
P34B_оценен посредством (was assessed by)
P41B_был классифицирован (was classified by)
P131B_идентифицирует (identifies)
P78B_идентифицирует (identifies)
P129B_является темой для (is subject of)
P17F_обусловлено (was motivated by)
P134B_была продолжена (was continued by)
P28F_опеку отдал (custody surrendered by)
P70F_документирует (documents)
P79_начало ограничено (beginning is qualified by)
P65F_показывает визуальный предмет (shows visual item)
P134F_продолжила (continued)
P73B_является переводом (is translation of)
P35B_идентифицировано посредством (identified by)
P25B_переместило (moved by)
P99B_была распущена (was dissolved by)
P135F_создало тип (created type)
P96B_дала рождение (gave birth)
```

Фрагменты реализации онтологии CRM на языке OWL

1. Определение класса E35_ через его надклассы E41_ и E33_:

```
<owl:Class rdf:ID="E35_">
    <rdfs:subClassOf rdf:resource="#E41_"/>
    <rdfs:subClassOf rdf:resource="#E33_"/>
  </owl:Class>
```

2. Определение свойства P84B_, его домена (E54_), диапазона (E52_) и обратного свойства (P84F_):

```
<owl:ObjectProperty rdf:about="#P84B_">
  <rdfs:domain rdf:resource="#E54_"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#P84F_"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#E52_"/>
</owl:ObjectProperty>
```

3. Определение свойства P99B_, его домена (E74_), диапазона (E68_) надсвойства (P11B_), обратного свойства (P99F_) и обратного однозначного свойства (P93B_):

```
<owl:ObjectProperty rdf:ID="P99B_">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="P99F_"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#E74_"/>
    <rdfs:subPropertyOf>
      <owl:ObjectProperty rdf:ID="P11B_"/>
    </rdfs:subPropertyOf>
    <rdfs:range rdf:resource="#E68_"/>
    <rdfs:subPropertyOf>
      <owl:InverseFunctionalProperty rdf:ID="P93B_"/>
    </rdfs:subPropertyOf>
  </owl:ObjectProperty>
```

Приложение 2. Иерархия классов онтологии вин

Краткое описание

Рис. П2-1. Иерархия онтологии вин.

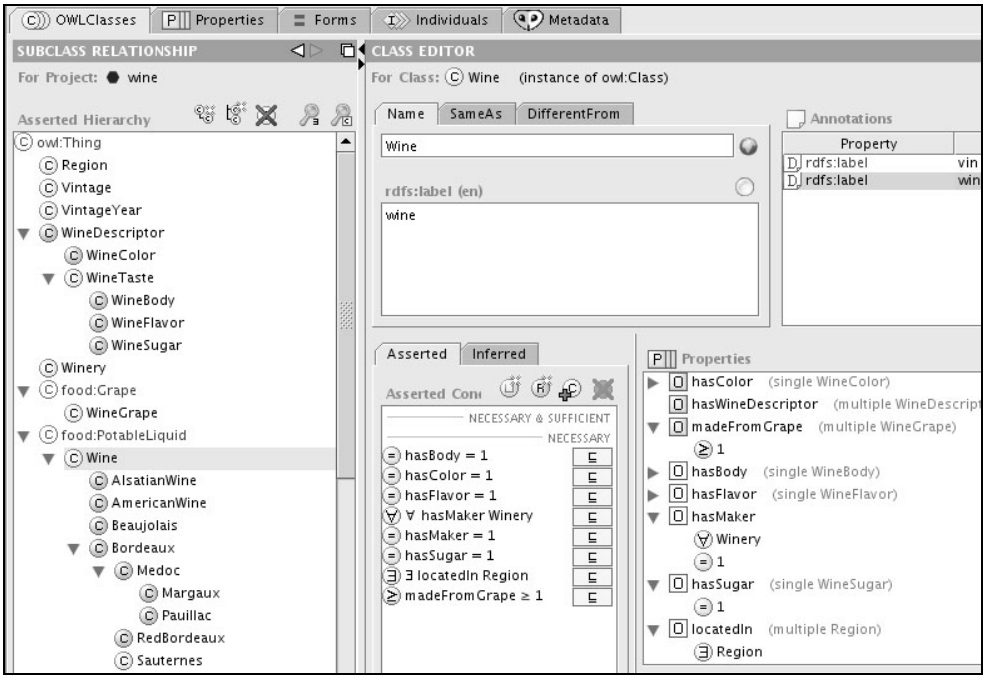

Рис. П2-2. Класс «Вино» и список ограничений, накладываемых на класс

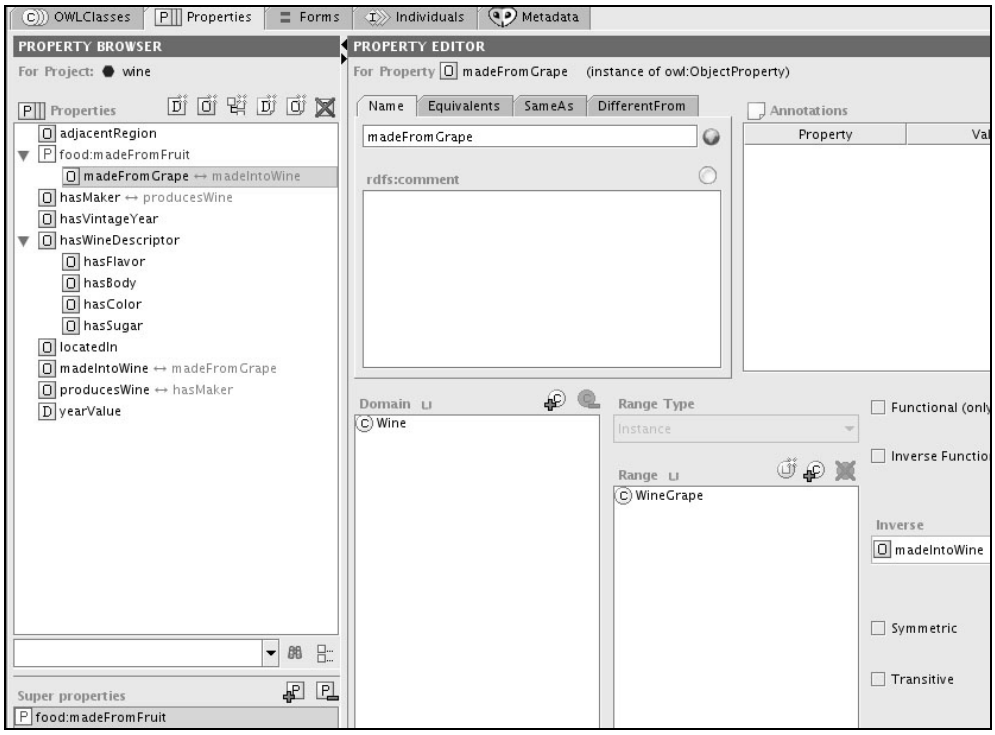

Рис. П2-3. Свойства онтологии вин.
Свойство madeFromGrape (сделаноИзВинограда) связывает домен Wine (Вино) и диапазон WineGrape (Виноград)

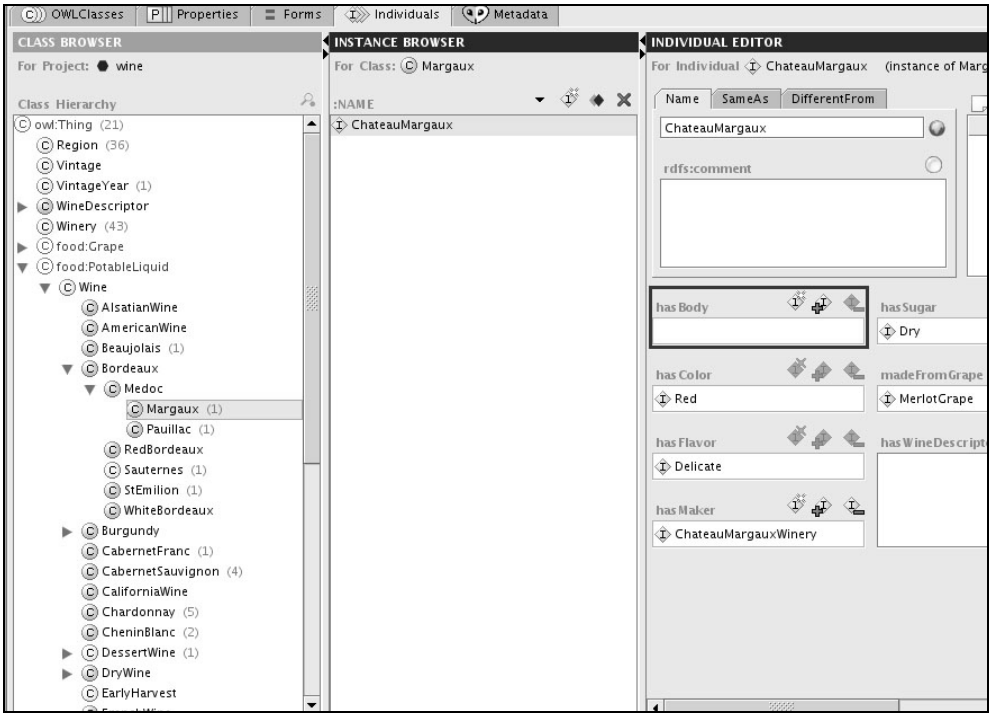

Рис. П2-4. Индивиды онтологии вин

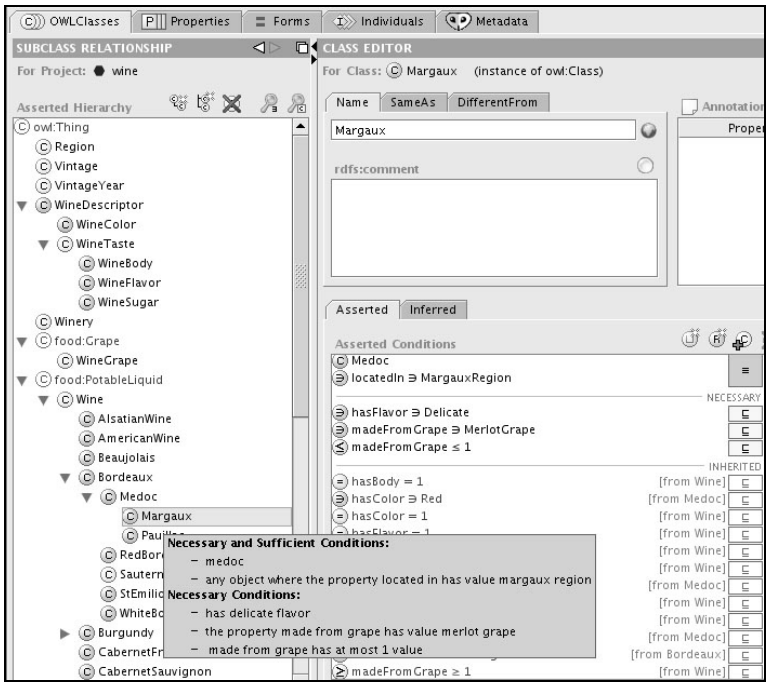

Рис. П2-5. Класс «Margaux» (Марго). Унаследованные и новые ограничения

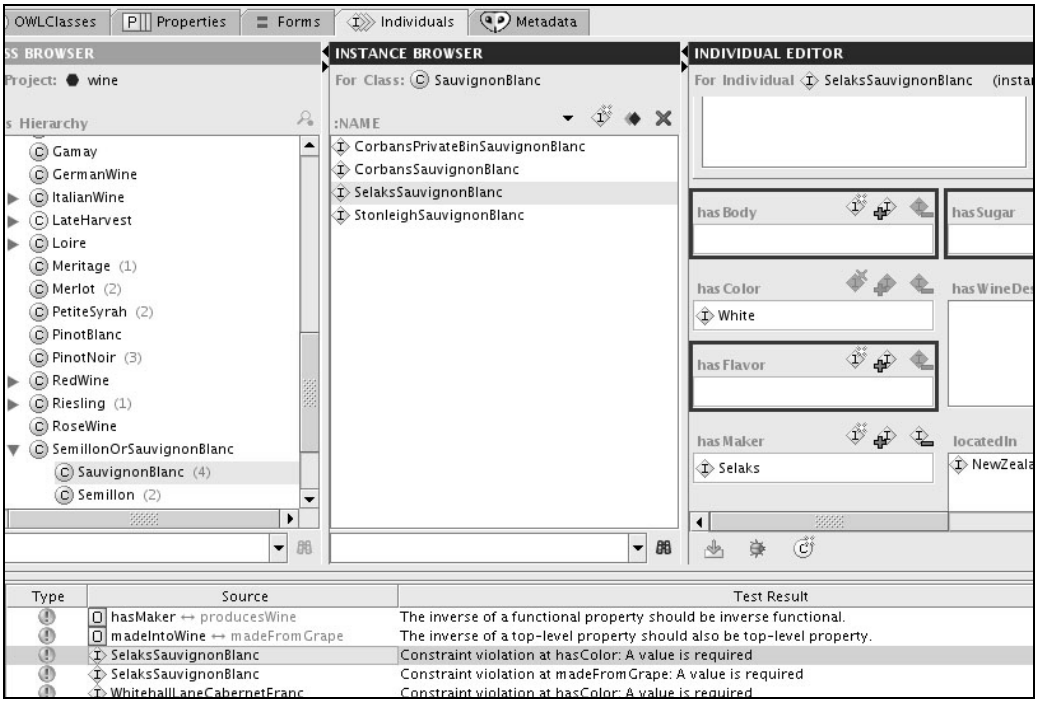

Рис. П2-6. Результаты выполнения тестов проверки целостности онтологии. Обязательное свойство hasColor (имеетЦвет) для индивида SelaksSauvignonBlanc не задано

Иерархия Классов

```
owl:Thing
 Region
 Vintage
 VintageYear
 WineDescriptor
  WineColor
  WineTaste
   WineBody
   WineFlavor
   WineSugar
 Winery
 food:Grape
  WineGrape
 food:PotableLiquid
  Wine
   AlsatianWine
   AmericanWine
   Beaujolais
   Bordeaux
    Medoc
     Margaux
     Pauillac
    RedBordeaux
    Sauternes
    StEmilion
    WhiteBordeaux
   Burgundy
    RedBurgundy
     CotesDOr
    WhiteBurgundy
     Meursault
   CabernetFranc
   CabernetSauvignon
   CaliforniaWine
   Chardonnay
   CheninBlanc
   DessertWine
    IceWine
    SweetRiesling
   DryWine
    DryRedWine
    DryWhiteWine
   EarlyHarvest
   FrenchWine
   FullBodiedWine
   Gamay
```

```
GermanWine
ItalianWine
 Chianti
LateHarvest
 (IceWine)...
 (Sauternes)...
Loire
 Anjou
 Muscadet
 Sancerre
 Tours
 WhiteLoire
Meritage
Merlot
PetiteSyrah
PinotBlanc
PinotNoir
RedWine
 (DryRedWine)...
 Port
 (RedBordeaux)...
 (RedBurgundy)...
Riesling
 DryRiesling
 (SweetRiesling)...
RoseWine
SemillonOrSauvignonBlanc
 SauvignonBlanc
 Semillon
SweetWine
TableWine
 RedTableWine
 WhiteTableWine
TexasWine
WhiteWine
 (DryWhiteWine)...
 (WhiteBordeaux)...
 (WhiteBurgundy)...
 (WhiteLoire)...
 WhiteNonSweetWine
Zinfandel
```

ОБ АВТОРАХ

СОЛОВЬЕВ Валерий Дмитриевич
д-р физ.-мат. наук, профессор
Кафедра теоретической кибернетики
Факультет вычислительной математики и кибернетики
Казанский государственный университет

ДОБРОВ Борис Викторович
канд. физ.-мат. наук
НИВЦ МГУ им. М.В. Ломоносова

ИВАНОВ Владимир Владимирович
мнс
Институт информатики
Факультет вычислительной математики и кибернетики
Казанский государственный университет

ЛУКАШЕВИЧ Наталья Валентиновна
канд. физ.-мат. наук
НИВЦ МГУ им. М.В. Ломоносова